

MENU

SEARCH

INDEX

DETAIL

JAPANESE

BACK

NEXT

8 / 18

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-172657
 (43)Date of publication of application : 23.06.2000

(51)Int.Cl. G06F 15/177
 G06F 13/00
 G06F 15/16

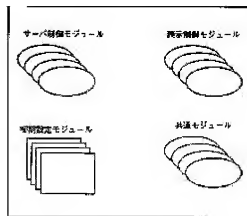
(21)Application number : 10-349067 (71)Applicant : FUJITSU LTD
 (22)Date of filing : 08.12.1998 (72)Inventor : SEGUCHI YOSHIYUKI
 YAMASHIMA HIROYUKI

(54) SYSTEM AND METHOD FOR DISTRIBUTED PROCESSING, COMPUTER- READABLE RECORDING MEDIUM WITH PROGRAM FOR COMPUTER TO EXECUTE THE SAME METHOD RECORDED THEREIN, SERVER DEVICE AND CLIENT DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a distributed processing system capable of comfortably utilizing a service from a server regardless of difference in the classes of clients.

SOLUTION: A service program for executing service is divided into a server control module required for executing the service on the side of server, a display control module for controlling display on a display device provided to a client, an initial setting module prepared for each client class so as to define a method for acquiring the required service module from the server to the client and a common module with which an execution position does not depend on the client and the server. When the resource lack of the client is detected in the case of executing this service program, the partial activation of the common module is requested to the server, and when the resource lack of the server is detected, the partial activation of the common module is requested to the other server.



LEGAL STATUS

[Date of request for examination] 29.10.2001
 [Date of sending the examiner's decision of rejection] 10.05.2005
 [Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(51) Int.Cl. ⁷	識別記号	F I	テレポート* (参考)	
G 0 6 F	15/177	G 0 6 F	15/177	6 7 4 A 5 B 0 4 5
	13/00		13/00	3 5 7 Z 5 B 0 8 9
	15/16		15/16	6 2 0 B

審査請求 未請求 請求項の数25 O L (全 35 頁)

(21) 出願番号 特願平10-349067

(22) 出願日 平成10年12月8日 (1998.12.8)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(72) 発明者 瀬口 義之

福岡県福岡市早良区百道浜2丁目2番1号
富士通九州通信システム株式会社内

(72) 発明者 山崎 弘之

神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

(74) 代理人 100089118

弁理士 酒井 宏明

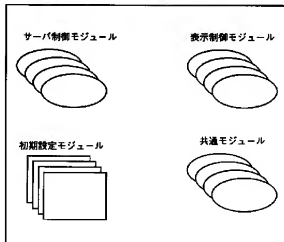
最終頁に続く

(54) 【発明の名称】 分散処理システム、分散処理方法、その方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体、サーバ装置およびクライアント装置

(57) 【要約】

【課題】 クライアントの種別の差異にかかわらずサーバからのサービスを快適に利用できる分散処理システムを提供すること。

【解決手段】 サービス実行をおこなうサービスプログラムを、サーバ側でサービス実行をおこなうために必要なサーバ制御モジュール、クライアントに備えられた表示装置に対する表示制御をおこなう表示制御モジュール、クライアント種別毎に用意されかつクライアントに必要なサービスモジュールのサーバからの取得方法を定義する初期設定モジュール、実行位置をクライアントとサーバとに依存しない共通モジュールに分割する。このサービスプログラムの実行において、クライアントのリソース不足が検出された場合は、共通モジュールの一部の起動をサーバに依頼し、サーバのリソース不足が検出された場合は、共通モジュールの一部の起動を他のサーバに依頼する。



【特許請求の範囲】

【請求項1】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおいて、前記サーバ装置は、サービスの提供および利用をおこなうための個別に実行可能な複数のモジュールと、前記モジュールの一種であり、前記クライアント装置がサービスを利用するのに必要なモジュールを該クライアント装置側で取得するための初期設定モジュールと、前記複数のモジュールを実行することでサービスの提供をおこなうサーバ提供手段と、

を備え、前記クライアント装置は、前記サーバ装置から前記初期設定モジュールを取得し、取得した初期設定モジュールの実行によって前記サーバ装置から前記必要なモジュールを取得し、取得したモジュールを実行することでサービスの利用をおこなうサービス利用手段を備えたことを特徴とする分散処理システム。

【請求項2】 前記サーバ装置は、前記モジュールの一種として前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを備え、前記サービス提供手段は、前記共通モジュールを実行している状態において、前記サーバ装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、同機能を有する他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求することを特徴とする請求項1に記載の分散処理システム。

【請求項3】 前記サーバ装置は、前記モジュールの一種として前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを備え、前記サービス利用手段は、前記共通モジュールを実行している状態において、前記クライアント装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、前記サーバ装置に対して該サーバ装置側での前記共通モジュールの起動を要求することを特徴とする請求項1または2に記載の分散処理システム。

【請求項4】 前記サーバ装置は、前記モジュールが更新された際に、該モジュールを配信した先のクライアント装置に、前記更新されたモジュールの更新を通知する更新通知手段を備え、

前記クライアント装置は、前記更新されたモジュールの更新が通知された場合に、前記サーバ装置から前記更新されたモジュールを取得する更新モジュール取得手段を備えたことを特徴とする請求項1～3のいずれか一つに記載のサーバ装置。

【請求項5】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法において、

前記サーバ装置が、前記クライアント装置の要求に応じて、サービスを利用するのに必要なモジュールを該クライアント装置側で取得するための初期設定モジュールを配信する初期設定モジュール配信ステップと、前記クライアント装置が、前記初期設定モジュールを取得し、取得した初期設定モジュールを実行して、前記サーバ装置から前記必要なモジュールを取得し、取得したモジュールを実行する初期設定モジュール実行ステップと、を含むことを特徴とする分散処理方法。

【請求項6】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法において、前記サーバ装置が、前記サーバ装置と他のサーバ装置との双方の実行環境において実行可能な共通モジュールを実行するサーバ共通モジュール実行ステップと、前記サーバ装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、前記他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップと、を含んだことを特徴とする分散処理方法。

【請求項7】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法において、前記クライアント装置が、前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを実行するクライアント共通モジュール実行ステップと、前記クライアント装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、前記サーバ装置に対して該サーバ装置側での前記共通モジュールの起動を要求するクライアント共通モジュール起動要求ステップと、を含んだことを特徴とする分散処理方法。

【請求項8】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法において、前記サーバ装置が、前記クライアント装置の要求に応じて、サービスを利用するのに必要なモジュールを配信した際に、配信先のクライアント装置のアドレスをモジュール配信先テーブルに登録するモジュール配信先登録ステップと、前記モジュールが更新された際に、前記モジュール配信先テーブルに登録されたクライアント装置のアドレス先に、前記更新されたモジュールの更新を通知するモジュール更新通知ステップと、前記クライアント装置が、前記更新されたモジュールの更新の通知を受けた場合に、前記サーバ装置から前記更新されたモジュールを取得する更新モジュール取得ステ

ップと、

を含んだことを特徴とする分散処理方法。

【請求項 9】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

前記サーバ装置から、前記クライアント装置の要求に応じて前記クライアント装置においてサービスを利用するのに必要なモジュールを該クライアント装置側で取得する

ための初期設定モジュールを配信する初期設定モジュール配信ステップと、

前記クライアント装置において前記初期設定モジュールを取得し、取得した初期設定モジュールを実行すること

で前記サーバ装置から前記必要なモジュールを取得し、取得したモジュールを実行する初期設定モジュール実行ステップと、

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 10】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを前記サーバ装置において実行するサーバ共通モジュール実行ステップと、

前記サーバ装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップと、

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 11】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを前記クライアント装置において実行するクライアント共通モジュール実行ステップと、

前記クライアント装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、前記サーバ装置に対して該サーバ装置側での前記共通モジュールの起動を要求するクライアント共通モジュール起動要求ステップと、

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 12】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

前記サーバ装置から、前記クライアント装置においてサービスを利用するのに必要なモジュールを配信する際に、配信先のクライアント装置のアドレスをモジュール配信先テーブルに登録するモジュール配信先登録ステップと、

前記モジュールが更新された際に、前記モジュール配信先テーブルに登録されたクライアント装置のアドレス先に、前記更新されたモジュールの更新を通知するモジュール更新通知ステップと、

前記クライアント装置に、前記更新されたモジュールの取得の指示が通知された場合に、前記サーバ装置から前記更新されたモジュールを取得させる更新モジュール取得ステップと、

を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 13】 ネットワークに接続されてクライアント装置とともに分散処理システムを構築するサーバ装置において、

サービスの提供および利用をおこなうための個別に実行可能な複数のモジュールと、

前記モジュールの一種であり、前記クライアント装置がサービスを利用するのに必要なモジュールを該クライアント装置側で取得するための初期設定モジュールと、

前記クライアント装置からの前記初期設定モジュールの取得要求に応じて該クライアント装置の種別に応じた初期設定モジュールを配信し、該初期設定モジュールが前記クライアント装置において実行されることによって要求されるモジュールを配信し、前記複数のモジュールを実行することでサービスの提供をおこなうサービス提供手段と、

を備えたことを特徴とするサーバ装置。

【請求項 14】 前記モジュールの一種として前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを備え、

前記サービス提供手段は、前記サーバ装置の処理能力や負荷状態の情報を一定時間毎に取得するリソース監視機構を設け、前記共通モジュールを実行している状態において、前記リソース監視機構によって取得された情報が処理能力の低下や負荷の増大等によるリソース不足状態を示す場合に、同機能を有する他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求することを特徴とする請求項 13 に記載のサーバ装置。

【請求項 15】 前記サービス提供手段は、複数の前記共通モジュールを実行している状態において、前記サー

バ装置のリソース不足状態が解消されるまで、起動要求の対象となる共通モジュールの選択をおこない、前記他のサーバ装置に対して前記選択した共通モジュールの起動の要求を繰り返すことを特徴とする請求項14に記載のサーバ装置。

【請求項16】 前記サービス提供手段は、前記他のサーバ装置から前記共通モジュールの起動を要求された場合に、前記共通モジュールを起動し、前記他のサーバ装置または前記クライアント装置からの前記起動した共通モジュールへのアクセスを許可することを特徴とする請求項14または15に記載のサーバ装置。

【請求項17】 前記共通モジュールの起動をおこなう前記サーバ装置の実行位置情報を登録するモジュール実行位置テーブルを備え、前記サービス提供手段は、前記モジュール実行位置テーブルに登録された前記実行位置情報に基づいて前記共通モジュールにアクセスすることを特徴とする請求項14～16のいずれか一つに記載のサーバ装置。

【請求項18】 前記モジュールを配信した先のクライアント装置のアドレスを登録するモジュール配信先テーブルと、前記サーバ装置において前記モジュールが更新された際に、前記モジュール配信先テーブルに登録されたクライアント装置のアドレス先に前記更新されたモジュールの更新を通知する更新通知手段と、を備えたことを特徴とする請求項13～17のいずれか一つに記載のサーバ装置。

【請求項19】 ネットワークに接続されてサーバ装置とともに分散処理システムを構築するクライアント装置において、前記サーバ装置が提供するサービスを利用するのに必要なモジュールを前記サーバ装置から取得するための初期設定モジュールを前記サーバ装置から取得し、取得した初期設定モジュールの実行によって必要なモジュールを取得し、取得したモジュールを実行することでサービスの利用をおこなうサービス利用手段を備えたことを特徴とするクライアント装置。

【請求項20】 前記初期設定モジュールは、前記クライアント装置の処理能力や負荷状態の調査をおこない、該調査結果に応じて、前記サーバ装置が提供するサービスをj 40
利用するのに必要なモジュールのうち前記サーバ装置から取得するモジュールを選択することを特徴とする請求項19に記載のクライアント装置。

【請求項21】 前記サーバ装置から取得したモジュールを管理するモジュール管理テーブルを備え、前記サービス利用手段は、前記モジュール管理テーブルにサービスを利用するのに必要なモジュールが登録されていない場合に、該モジュールを前記サーバ装置から取得し、取得したモジュールを前記モジュール管理テーブルに登録することを特徴とする請求項19または20に

記載のクライアント装置。

【請求項22】 前記サービス利用手段は、サービスを利用するのに必要なモジュールの一種として前記サーバ装置と前記クライアント装置との双方の実行環境において実行可能な共通モジュールを前記サーバ装置から取得し、前記クライアント装置の処理能力や負荷状態の情報を一定時間毎に取得するリソース監視機構を設け、前記共通モジュールを実行している状態において、前記リソース監視機構によって取得された情報が処理能力の低下や負荷の増大等によるリソース不足状態を示す場合に、前記サーバ装置に対して該サーバ装置側での前記共通モジュールの起動を要求することを特徴とする請求項19～21のいずれか一つに記載のクライアント装置。

【請求項23】 前記サービス利用手段は、複数の前記共通モジュールを実行している状態において、前記クライアント装置のリソース不足状態が解消されるまで、起動要求の対象となる共通モジュールの選択をおこない、前記サーバ装置に対して前記選択した共通モジュールの起動の要求を繰り返すことを特徴とする請求項22に記載のクライアント装置。

【請求項24】 前記共通モジュールの起動をおこなう前記サーバ装置の実行位置情報を登録するモジュール実行位置テーブルを備え、前記サービス利用手段は、前記モジュール実行位置テーブルに登録された前記実行位置情報に基づいて前記共通モジュールにアクセスすることを特徴とする請求項22または23に記載のクライアント装置。

【請求項25】 前記サーバ装置から更新されたモジュールの更新の通知を受けた場合に、前記サーバ装置から前記更新されたモジュールを取得する更新モジュール取得手段を備えたことを特徴とする請求項19～24のいずれか一つに記載のクライアント装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワークを介して接続されたサーバおよびクライアントから構成され、ユーザにサービスを提供する分散処理システム、分散処理方法、その方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体、サーバ装置およびクライアント装置において、特に、サービスを実現するために必要なサービスプログラムを実行する際のサーバとクライアントの双方における負荷とサービスプログラムの保守作業とを低減させる分散処理システム、分散処理方法、その方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体、サーバ装置およびクライアント装置に関する。

【0002】

【従来の技術】近年、通信インフラストラクチャの整備

の向上と、LAN (Local Area Network) やインターネットの普及により、ネットワークを介して所望のサービスを受けることを可能とする分散処理システムの利用が一般的になっている。特に、ユーザの利用する複数のクライアント装置 (以下、単にクライアントと称する) と、サービスを提供するサーバ装置 (以下、単にサーバと称する) とから構成され、クライアントがサーバにアクセスすることによって所望のサービスを受けることのできるクライアント・サーバ方式は、ネットワーク上において種々のサービス資源を共有または管理する上での最適なネットワーク接続方式として確立されたものとなっている。

【0003】一方、LANからWAN (Wide Area Network) へ、さらにインターネットへ、と地域または社内において点在したネットワーク間を相互接続することによって、ユーザすなわちクライアントが接続できるネットワークの規模が拡大している。さらに、ネットワークを利用するユーザの数、すなわち各ネットワーク内に所属するクライアントの数が増大しており、上記したネットワークの規模の拡大にともなって、サーバの負荷および通信トラフィックの増大が問題となっている。

【0004】このため、現在においては、小規模なネットワーク内においても、提供されるサービス種に応じて複数のサーバを用意し、サービス実行における負荷の分散がおこなわれる傾向がある。例えば、データベースサービスを提供するデータベースサーバ、ファイル一括管理するファイルサーバ、電子メールを管理するメールサーバ、大規模かつ複雑な計算処理を専門とする計算処理サーバ等のサーバがネットワーク上に分散して配置される。

【0005】このようにネットワーク上のサービス授受に関するソフトウェアの進展が見られる一方で、ネットワークに接続される装置、特にクライアントとなりうる多種のハードウェアに対する進展が望まれる。クライアントは、主に、オペレーティングシステムや、種々のアプリケーションプログラムを実行するCPU (Central Processing Unit) と、ネットワークに接続するための通信インターフェースと、CPUによって実行されるプログラムやデータを記憶する記憶装置と、キーボードやマウス等の入力装置と、CRTや液晶ディスプレイ等の表示装置と、から構成される。サーバもまた、クライアントと同様な構成であるが、さらに、各種サービスを提供するための周辺装置が接続される場合がある。

【0006】クライアントの構成は、CPUの種類によってその仕様が差別されており、多くの場合、このCPUに適したオペレーティングシステムを採用することでスタンダードとして動作することが可能である。

【0007】したがって、異なるハードウェア仕様の

クライアント間においても、ネットワークに接続して共有資源の活用を図ろうとする要望が生じてくる。すなわち、どのようなハードウェア仕様のクライアントからであっても、サーバからの同様なサービスを受けたいというユーザの要求が高まっている。この要求に応えるためには、ネットワークで採用されている通信プロトコルと同一の通信プロトコルを使用した通信をおこなうこと、すなわち通信インターフェースを共通にすることに加えて、サービスを提供するサーバにアクセスするためのサービスプログラムが必要になる。このサービスプログラムは、クライアント上で実行されるため、クライアントのハードウェア仕様、特にオペレーティングシステムに応じて個々に用意する必要がある。

【0008】従来の分散型グループウェアや分散型ネットワークサービスなどは、一般的に、サービスに必要な機能は上記したサービスプログラムとしてあらかじめサーバおよびクライアントに登録されており、このサービスプログラムが、異種のハードウェア仕様または異種のオペレーティングシステム間のインターフェースの違いを吸収して、サーバからクライアントへのサービス提供を実現している。

【0009】以下に、クライアントがサーバからサービスを受ける例として、クライアントがサーバのデータベースから所望のデータを取得する場合について説明する。この場合、アクセス先となるサーバは、大容量のハードディスク等の記憶装置を備え、この記憶装置の記憶領域上にデータベースを構築しているデータベースサーバである。なお、クライアントは、このデータベースサーバにアクセスし、所望のデータを検索して取得するためのサービスプログラム (以下、検索プログラムと称する) をあらかじめ搭載している。

【0010】まず、クライアントは、上記した検索プログラムを起動する。なお、この起動段階において、すでにデータベースサーバとの接続が確立されたものとする。検索プログラムは、一般に、検索条件 (クエリ) の入力を受け付けるクエリ入力処理と、サーバから転送される検索結果を表示する結果表示処理を含んでいる。ユーザは、クライアントに備えられた入力装置を使用して、欲するデータに関するキーワードの入力や検索プログラムによって提示されたカテゴリの選択を進めることによって、クエリの入力を完了させる。

【0011】検索プログラムは、クエリ入力処理が完了すると、入力されたクエリの情報をデータベースサーバに送信して、データの検索および転送を要求する。データベースサーバは、クライアントの検索プログラムに対応するサービスプログラムがデモンとして実行されており、クライアントからのクエリ情報を受信すると直ちに、このクエリ情報に基づいて、サーバの記憶装置に蓄積された膨大なデータからユーザの欲するデータを特定する。ついで、データベースサーバは、特

定したデータを、要求のあったクライアントに向けて送信する。

【0012】そして、クライアントの検索プログラムは、データベースサーバから送信されてきたデータを受信し、このデータを表示装置上に表示する。これにより、ユーザは、所望のデータを取得することができる。このように、クライアントは、検索プログラムという比較的軽量のプログラムをサーバプログラムとして搭載することのみで、膨大な容量を必要とするデータベースを利用することができる。

【0013】また、従来における分散処理システムの他の例として、特開平10-21081号公報に開示のクライアント・サーバシステムは、クライアントにおいて実行されるアプリケーションプログラム（サービスプログラム）を、個別に実行可能な複数のモジュールに分割し、この複数のモジュールから構成されるアプリケーションプログラムをサーバの記憶装置上に保存する。そして、クライアントがこのサーバにアクセスしてサービスを利用したい場合には、サーバに保存されたアプリケーションプログラムのうち、実行に必要な部分のモジュールのみをクライアントの記憶装置にダウンロードし、このモジュールを実行する。

【0014】実行中のモジュールにおいて、さらに他のモジュールの実行が必要になる場合には、そのモジュールをサーバからダウンロードする。この際、クライアントにおいてダウンロードされたモジュールの総容量が所定の容量を超える場合には、超過した分のモジュールをクライアントの記憶装置から削除する。これにより、クライアントにおいて、大容量のアプリケーションプログラムを実行する必要がある場合にも、クライアントの記憶装置等の資源を圧迫することがなくなる。

【0015】よって、以上に説明した従来の分散処理システムにおいては、ネットワーク上の共有資源を利用することができるとともに、多種多様な様子のクライアント間においても、通信インターフェースを共通にすることによりネットワークへの接続が確立でき、サーバが提供するサービスを利用できる。

【0016】【発明が解決しようとする課題】しかしながら、従来の分散処理システムでは、サーバおよびクライアントに登録されたサービスプログラムは固定的であり、サービスの一部機能の追加や変更をおこなう場合はサーバやクライアントに登録されたサービスプログラム全体の変更をおこなう必要があり、効率的な保守をおこなうことができなかった。

【0017】また、サービスプログラムは一般的にサーバ用プログラムとクライアント用プログラムとして明確に区別されているため、クライアントやサーバの処理能力や機能に応じて、サービス処理の実行位置を移行し負

荷の分散をおこなうことは困難であった。

【0018】また、従来の分散処理システムでは、ユーザが、クライアントに登録するサービスプログラムをクライアントの資源や機能に応じて選択することができるが、これはクライアント毎に実現可能なサービスを制限するものであり、サーバの提供するすべてのサービスを利用できないという、クライアントに依存したサービス制限の問題があった。

【0019】さらに、従来の分散処理システムでは、一般的に、ユーザまたはシステム管理者が、クライアント毎に、そのクライアントの機能に応じたサービスプログラムをあらかじめインストールしなければならない、ユーザまたはシステム管理者の負担を大きくする傾向があった。

【0020】また、上記したインストールの必要性に関連して、サービスプログラムに対する機能追加や変更が行われる際にも、サーバおよびクライアント装置のサービスプログラム全体を再インストールする必要があり、サービスプログラム内の個別の機能に対する保守は困難であった。さらに、すべてのユーザがサービスプログラムの更新時期を把握することが困難であるという問題があった。

【0021】また、従来の分散型グループウェアや分散型ネットワークサービスでは、一つのクライアント装置で複数のユーザを対象としたマルチユーザシステムを採用している場合が多いが、ユーザ毎に利用するサービスの機能が異なる場合であっても、全ての機能をサービスプログラムとしてクライアントに保持する必要があり、必ずしも効率的にクライアントの資源を利用したものはなかった。

【0022】特開平10-21081号公報に開示のクライアント・サーバシステムは、サービスプログラムをモジュール化して実行に必要なモジュールのみがクライアントにダウンロードされるので、上述したサービスプログラムのインストール作業の問題、サービスプログラム内の個別の機能に対する保守の問題、クライアントの資源の問題を解決するが、サービスプログラムの更新時期を迅速に把握できるものではなかった。

【0023】また、このクライアント・サーバシステムにおいては、クライアントがサービスを受ける際には、サーバからのモジュールのダウンロードを必ず実行する必要がある。したがって、クライアントのCPUがモジュールを実行できる状態までに、少なくともこのダウンロードのための時間が必要とされ、新規なモジュールをつぎつぎと呼び出すようなサービスプログラムを実行する場合には、スループットが低下し、クライアントの負荷が増大するだけでなく、サーバの負荷および通信トラヒックの増大を招きうる。

【0024】さらに、このクライアント・サーバシステムにおいては、記憶装置のみに対する資源の有効活用を

11

可能としており、クライアントが同時に複数のサーバにアクセスする場合等におけるCPUに対する負荷、またはCPUの実行対象位置となる揮発性メモリの圧迫を考慮したのではなく、必ずしも他のサーバやクライアントを含むネットワーク全体のパフォーマンスを向上させるものではなかった。

【0025】本発明は、上記に鑑みてなされたものであって、クライアントの種類の差異にかかわらずサーバからのサービスを快適に利用できる分散処理システムを提供することを目的とする。

【0026】

【課題を解決するための手段】上述した課題を解決し、目的を達成するために、請求項1の発明において、ネットワークを介して互いに接続されたサーバ装置（サーバ）とクライアント装置（クライアント）とを備えた分散処理システムは、サーバが、サービスの提供および利用をおこなうための個別に実行可能な複数のモジュール（サービスモジュール）と、サービスモジュールの一種であり、クライアントがサービスを利用するのに必要なサービスモジュール（表示制御モジュール等）をそのクライアント側で取得するための初期設定モジュールと、複数のサービスモジュール（サーバ側制御モジュール等）を実行することでサービスの提供をおこなうサービス提供手段（後述する「サーバ側サービス起動処理」と、「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるサーバ側処理に相当）と、を備え、クライアントが、サーバから初期設定モジュールを取得し、取得した初期設定モジュールの実行によってサーバから必要なサービスモジュールを取得し、取得したモジュールを実行することでサービスの利用をおこなうサービス利用手段（後述する「クライアント側サービス起動処理」と、「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるクライアント側処理に相当）を備えたことを特徴とする。

【0027】この請求項1の発明によれば、サーバは、クライアントの要求に応じて、クライアントの種類毎に異なり、かつサービスの実行に必要なモジュールをサーバから取得することのできる初期設定モジュールをクライアントに配信し、クライアントは、サーバからこの初期設定モジュールを取得して実行することによって、クライアントにおいて必要なサービスモジュールのみをサーバから取得することができ、この取得したサービスモジュールを実行することにより、サービスを利用することができ、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続された場合であっても、クライアントに応じた初期設定モジュールを取得するだけで、サーバによって提供されるサービスがクライアント間において差異なく利用でき、また、クライアントにおいてサービス利用手段により該当する初期設定モジュールを自動的に取得することができるの

12

で、ユーザは、クライアントの種別を意識することなく、サーバの提供するサービスを容易に利用することができる。

【0028】また、請求項2の発明において、請求項1に記載の分散処理システムは、サーバが、サービスモジュールの一種としてサーバとクライアントとの双方の実行環境において実行可能な共通モジュールを備え、さらに、サービス提供手段が、この共通モジュールを実行している状態（後述する「サービス起動処理」のステップS115s、「クライアントにおけるモジュール実行環境移行処理」のステップS123s、または「サーバにおけるモジュール実行環境移行処理」のステップS143sに相当）において、サーバが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（後述する「システム監視スレッドの処理」に相当）に、このサーバと同機能を有する他のサーバに対してその他のサーバ側での共通モジュールの起動を要求する（「サーバにおけるモジュール実行環境移行処理」のステップS145sに相当）ことを特徴とする。

【0029】この請求項2の発明によれば、サーバに、実行環境を問わない共通モジュールが備えられることで、サーバのリソース不足状態に対して、本来、このサーバにおいて実行すべき共通モジュールを、他のサーバにおいて実行するように実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、多くのクライアントのアクセスにより、サーバの資源が圧迫された場合に、サーバにおいて実行されていた共通モジュールを、リソースに余裕のある他のサーバにおいて実行できるようになり、サーバのリソース不足によるスループットの低下を回避することができる。

【0030】また、請求項3の発明において、請求項1または2に記載の分散処理システムは、サーバが、サービスモジュールの一種としてサーバとクライアントとの双方の実行環境において実行可能な共通モジュールを備え、さらに、クライアントのサービス利用手段が、この共通モジュールをクライアント側で実行している状態（「サービス起動処理」のステップS112cに相当）において、クライアントが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、サーバに対してそのサーバ側での共通モジュールの起動を要求する（「クライアントにおけるモジュール実行環境移行処理」のステップS124cに相当）ことを特徴とする。

【0031】この請求項3の発明によれば、サーバにサーバとクライアントとの双方の実行環境において実行可能な共通モジュールが備えられることで、クライアントのリソース不足状態に対して、本来、クライアントにおいて実行すべき共通モジュールを、サーバにおいて実行するように実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、他のアプリケー

13

ジョンプログラム等の実行により、クライアントの資源が圧迫された場合に、クライアントにおいて実行されていた共通モジュールをサーバにおいて実行できるようになり、クライアントのリソース不足によるスループットの低下を回避することができる。

【0032】また、請求項4の発明において、請求項1～3のいずれか一つに記載の分散処理システムは、サーバが、サービスモジュールの更新が生じた際に、そのサービスモジュールを配信した先のクライアントに、更新されたサービスモジュールの更新を通知する更新通知手段（後述する「サービスモジュール更新処理」のサーバ側処理に相当）を備え、クライアントは、サーバから更新されたサービスモジュールの更新が通知された場合に、更新されたサービスモジュールをサーバから取得する更新モジュール取得手段（「サービスモジュール更新処理」のクライアント側処理に相当）を備えたことを特徴とする。

【0033】この請求項4の発明によれば、サーバが、クライアントに対してサービスモジュールの更新を通知することができ、さらに、その通知は、過去にそのサービスモジュールの配信をおこなった先のクライアントに対しておこなわれるので、クライアントにおいて、ユーザが、特にサーバにおけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。

【0034】また、請求項5の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおいて採用される分散処理方法は、サーバが、クライアントの要求に応じて、サービスを利用するのに必要なモジュール（サービスモジュールであり、表示制御モジュールを含む）をそのクライアント側で取得するための初期設定モジュールを配信する初期設定モジュール配信ステップ（「初期設定モジュール取得処理」におけるサーバ側処理に相当）と、クライアントが、この初期設定モジュールを取得し、取得した初期設定モジュールを実行して、サーバから必要なサービスモジュールを取得し、取得したサービスモジュールを実行する初期設定モジュール実行ステップ（「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるクライアント側処理に相当）と、を含むことを特徴とする。

【0035】この請求項5の発明によれば、クライアントが、サーバから、クライアントの種別毎に異なり、かつサーバの実行に必要なサービスモジュールの取得をおこなう初期設定モジュールを取得して、これを実行することで、クライアントにおいて必要なサービスモジュールのみをサーバから取得することができ、この取得したサービスモジュールの実行により、サービスの利用を可能とするので、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続さ

14

れた場合であっても、ユーザは、サーバによって提供されるサービスを他のクライアントを利用した場合と同様に利用することができる。

【0036】また、請求項6の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおいて採用される分散処理方法は、サーバが、サーバと他のサーバとの双方の実行環境において実行可能な共通モジュールを実行するサーバ共通モジュール実行ステップ（「サービス起動処理」のステップS115s、「クライアントにおけるモジュール実行環境移行処理」のステップS123s、または「サーバにおけるモジュール実行環境移行処理」のステップS143sに相当）と、サーバが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、他のサーバに対してその他のサーバ側での共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップ（「サーバにおけるモジュール実行環境移行処理」のステップS145sに相当）と、を含んだことを特徴とする。

【0037】この請求項6の発明によれば、サーバにおいて、実行環境を問わない共通モジュールの実行中に、サーバのリソース不足状態が生じた際、この共通モジュールを、他のサーバにおいて実行するように要求して、共通モジュールの実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、多くのクライアントのアクセスにより、サーバの資源が圧迫された場合に、サーバにおいて実行されていた共通モジュールを、リソースに余裕のある他のサーバにおいて実行できるようになり、サーバのリソース不足によるスループットの低下を回避することができる。

【0038】また、請求項7の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおいて採用される分散処理方法は、クライアントが、サーバとクライアントとの双方の実行環境において実行可能な共通モジュールを実行するクライアント共通モジュール実行ステップ（「サービス起動処理」のステップS112cに相当）と、クライアントが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、サーバに対してそのサーバ側での共通モジュールの起動を要求するクライアント共通モジュール起動要求ステップ（「クライアントにおけるモジュール実行環境移行処理」のステップS124cに相当）と、を含んだことを特徴とする。

【0039】この請求項7の発明によれば、クライアントにおいて、実行環境を問わない共通モジュールの実行中に、クライアントのリソース不足状態が生じた際、この共通モジュールを、サーバにおいて実行するように要求して、共通モジュールの実行位置を移行することがで

15

きるので、たとえば、サービスプログラムの実行中に、他のアプリケーションプログラム等の実行により、クライアントの資源が圧迫された場合に、クライアントにおいて実行されていた共通モジュールをサーバにおいて実行できるように、クライアントのリソース不足によるスループットの低下を回避することができる。

【0040】また、請求項8の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおいて採用される分散処理方法は、サーバが、クライアントの要求に応じて、サーバを利用するのに必要なサービスモジュール（初期設定モジュール、表示制御モジュール等）を配信した際に、配信先のクライアントのアドレスをモジュール配信先テーブル（サーバデータ管理テーブル）に登録するモジュール配信先登録ステップ（「初期設定モジュール取得処理」のステップS93sおよび「初期設定モジュール処理」のステップS102sに相当）と、サービスモジュールが更新された際に、モジュール配信先テーブルに登録されたクライアントのアドレス先に、更新されたサービスモジュールの更新を通知するモジュール更新通知ステップ（「サービスモジュール更新処理」のサーバ側処理に相当）と、クライアントが、更新されたモジュールの更新の通知を受けた場合に、サーバから更新されたモジュールを取得する更新モジュール取得ステップ（「サービスモジュール更新処理」のクライアント側処理に相当）と、を含んだことを特徴とする。

【0041】この請求項8の発明によれば、サーバが、クライアントに対してサービスモジュールの更新を通知することができ、さらに、その通知は、モジュール配信先テーブルを参照することで、過去にそのサービスモジュールの配信をおこなった先のクライアントに対しておこなわれるので、クライアントにおいて、ユーザが、特にサーバにおけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。

【0042】また、請求項9の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバから、クライアントの要求に応じてそのクライアントにおいてサービスを利用するのに必要なモジュール（サービスモジュールであり、表示制御モジュール等）をそのクライアント側で取得するための初期設定モジュールを配信する初期設定モジュール配信ステップ（「初期設定モジュール取得処理」におけるサーバ側処理に相当）と、クライアントにおいて初期設定モジュールを取得し、取得した初期設定モジュールを実行することでサーバから必要なサービスモジュールを取得し、取得したモジュールを実行する初期設定モジュール実行ステップ（「初期設定モジュール

16

取得処理」および「初期設定モジュール処理」におけるクライアント側処理に相当）と、を実行させることを特徴とする。

【0043】この請求項9の発明によれば、この記録媒体をサーバおよびクライアントにインストールすることによって、クライアントが、サーバから、クライアントの種別毎に異なり、かつサービスの実行に必要なサービスモジュールの取得をおこなう初期設定モジュールを取得し、これを実行することで、クライアントにおいて必要なサービスモジュールのみをサーバから取得することができ、この取得したサービスモジュールの実行により、サービスの利用を可能とするプログラムを実行することができるので、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続された場合であっても、ユーザは、サーバによって提供されるサービスを他のクライアントを利用した場合と同様に利用することができる。

【0044】また、請求項10の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバとクライアントとの双方の実行環境において実行可能な共通モジュールをサーバにおいて実行するサーバ共通モジュール実行ステップ（「サービス起動処理」のステップS115s、「クライアントにおけるモジュール実行環境移行処理」のステップS123s、または「サーバにおけるモジュール実行環境移行処理」のステップS143sに相当）と、サーバが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、他のサーバに対してその他のサーバ側での共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップ（「サーバにおけるモジュール実行環境移行処理」のステップS145sに相当）と、を実行させることを特徴とする。

【0045】この請求項10の発明によれば、この記録媒体をサーバおよびクライアントにインストールすることによって、サーバにおいて、実行環境を問わない共通モジュールの実行中に、サーバのリソース不足状態が生じた際、この共通モジュールを、他のサーバにおいて実行するように要求して、共通モジュールの実行位置を移行することが可能なプログラムを実行することができるので、たとえば、サービスプログラムの実行中に、多くのクライアントのアクセスにより、サーバの資源が圧迫された場合に、サーバにおいて実行されていた共通モジュールを、リソースに余裕のある他のサーバにおいて実行できるように、サーバのリソース不足によるスループットの低下を回避することができる。

【0046】また、請求項11の発明において、ネットワークを介して互いに接続されたサーバとクライアント

17

とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバとクライアントとの双方の実行環境において実行可能な共通モジュールをクライアントにおいて実行するクライアント共通モジュール実行ステップ（「サービス起動処理」のステップS112cに相当）と、クライアントが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、サーバに対してそのサーバ側での共通モジュールの起動を要求するクライアント共通モジュール起動要求ステップ（「クライアントにおけるモジュール実行環境移行処理」のステップS124cに相当）と、を実行させることを特徴とする。

【0047】この請求項11の発明によれば、この記録媒体をサーバおよびクライアントにインストールすることによって、クライアントにおいて、実行環境を問わない共通モジュールの実行中に、クライアントのリソース不足状態が生じた際、この共通モジュールを、サーバにおいて実行するように要求して、共通モジュールの実行位置を移行することが可能なプログラムを実行することができるので、たとえば、サービスプログラムの実行中に、他のアプリケーションプログラム等の実行により、クライアントの資源が圧迫された場合に、クライアントにおいて実行されていた共通モジュールをサーバにおいて実行できるようになり、クライアントのリソース不足によるスループットの低下を回避することができる。

【0048】また、請求項12の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバから、クライアントにおいてサービスを利用するのに必要なサービスモジュール（初期設定モジュール、表示制御モジュール等）を配信した際に、配信先のクライアントのアドレスをモジュール配信先テーブル（サーバデータ管理テーブル）に登録するモジュール配信先登録ステップ（「初期設定モジュール取得処理」のステップS93sおよび「初期設定モジュール処理」のステップS102sに相当）と、サービスモジュールが更新された際に、モジュール配信先テーブルに登録されたクライアントのアドレス先に、更新されたサービスモジュールの更新を通知するモジュール更新通知ステップ（「サービスモジュール更新処理」のサーバ側処理に相当）と、クライアントに、更新されたサービスモジュールの取得の指示が通知された場合に、サーバから更新されたサービスモジュールを取得させる更新モジュール取得ステップ（「サービスモジュール更新処理」のクライアント側処理に相当）と、を実行させることを特徴とする。

【0049】この請求項12の発明によれば、この記録

18

媒体をサーバおよびクライアントにインストールすることによって、サーバが、クライアントに対してサービスモジュールの更新を通知することができ、さらに、その通知は、モジュール配信先テーブルを参照することで、過去にそのサービスモジュールの配信をおこなった先のクライアントに対しておこなうことが可能なプログラムを実行することができるので、クライアントにおいて、ユーザが、特にサーバにおけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。

【0050】また、請求項13において、ネットワークに接続されたクライアント装置（クライアント）とともに分散処理システムを構築するサーバ装置（サーバ）は、サービスの提供および利用をおこなうための個別に実行可能な複数のモジュール（サービスモジュール）と、サービスモジュールの一種であり、クライアントがサービスを利用するのに必要なサービスモジュール（表示制御モジュール等）をそのクライアント側で取得するための初期設定モジュールと、クライアントからの初期設定モジュールの取得要求に応じてそのクライアントの種別に応じた初期設定モジュールを配信し、その初期設定モジュールがクライアントにおいて実行されることによって要求されるサービスモジュールを配信し、サービスモジュールを実行することでサービスの提供をおこなうサービス提供手段（「サーバデーモン起動処理」と、「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるサーバ側処理に相当）と、を備えたことを特徴とする。

【0051】この請求項13の発明によれば、サーバは、クライアントの種別毎に異なるサービスモジュールの配信を可能とする初期設定モジュールをクライアントに配信するので、クライアント側でサービスを利用するためのサービスプログラムに必要なサービスモジュールの変更等が生じた場合にも、初期設定モジュールの内容を変更するだけで対応でき、クライアントの種別毎に存在する多くのサービスモジュールの管理が容易になる。

【0052】また、請求項14において、請求項13に記載のサーバ装置（サーバ）は、サービスモジュールの一種としてサーバとクライアントとの双方の実行環境において実行可能な共通モジュールを備え、サービス提供手段が、サーバの処理能力や負荷状態の情報を一定時間毎に取得するリソース監視機構（「システム監視スレッドの処理」に相当）を設け、共通モジュールを実行している状態において、リソース監視機構によって取得された情報が処理能力の低下や負荷の増大等によるリソース不足状態を示す場合に、このサーバと同機能を有する他のサーバに対してその他のサーバ側での共通モジュールの起動を要求する（「サーバにおけるモジュール実行環境移行処理」のステップS145sに相当）ことを特徴とする。

50

19

【0053】この請求項14の発明によれば、サーバに、実行環境を問わない共通モジュールが備えられることで、サーバのリソース不足状態に対して、本来、このサーバにおいて実行すべき共通モジュールを、他のサーバにおいて実行するように実行位置を移行することができるので、サーバのリソース不足が生じた場合に、サーバにおいて実行されていた共通モジュールを、リソースに余裕のある他のサーバにおいて実行できるようになり、サーバのスループットの低下を回避することができる。

【0054】また、請求項15において、請求項14に記載のサーバは、サービス提供手段が、複数の共通モジュールを実行している状態において、サーバのリソース不足状態が解消されるまで、起動要求の対象となる共通モジュールの選択をおこない、他のサーバに対して選択した共通モジュールの起動の要求を繰り返す（「サーバにおけるモジュール実行環境移行処理」に相当）ことを特徴とする。

【0055】この請求項15の発明によれば、実行されている複数の共通モジュールのうち、リソース不足が解消されるだけの共通モジュールを、他のサーバに対する起動要求の対象とするので、起動要求先のサーバのリソースへの大きな圧迫を回避することができる。

【0056】また、請求項16において、請求項14または15に記載のサーバは、サービス提供手段が、他のサーバから共通モジュールの起動要求された場合に、共通モジュールを起動し、他のサーバまたはクライアントからの起動した共通モジュールへのアクセスを許可する（「クライアントにおけるモジュール実行環境移行処理」および「サーバにおけるモジュール実行環境移行処理」に相当）ことを特徴とする。

【0057】この請求項16の発明によれば、起動要求された共通モジュールへのアクセスを、起動要求を発したサーバだけでなく、その共通モジュールの実行を必要とするクライアントに対しても直接に可能とするので、起動要求された共通モジュールの利用が容易に実現される。

【0058】また、請求項17において、請求項14～16のいずれか一つに記載のサーバは、共通モジュールの起動をおこなうサーバの実行位置情報を登録するモジュール実行位置テーブル（サーバデータ管理テーブル）を備え、サービス提供手段が、モジュール実行位置テーブルに登録された実行位置情報に基づいて共通モジュールにアクセスする（後述する「サービスモジュールアクセス処理」に相当）ことを特徴とする。

【0059】この請求項17の発明によれば、モジュール実行位置テーブルによって共通モジュールの実行位置を管理するので、このモジュール実行位置テーブルを参照することで、所望の共通モジュールのアクセスが容易になる。

20

【0060】また、請求項18において、請求項13～17のいずれか一つに記載のサーバは、サービスモジュールを配信した先のクライアントのアドレスを登録するモジュール配信先テーブル（サーバデータ管理テーブル）と、サーバにおいてサービスモジュールが更新された際に、モジュール配信先テーブルに登録されたクライアントのアドレス先に更新されたサービスモジュールの更新を通知する更新通知手段（「サービスモジュール更新処理」のサーバ側処理に相当）とを、備えたことを特徴とする。

【0061】この請求項18の発明によれば、モジュール配信先テーブルを参照することによって、更新されたサービスモジュールの取得が必要なクライアントのみに更新通知をおこなうので、効率的なサービスモジュールの更新処理が可能となる。

【0062】また、請求項19において、ネットワークに接続されてサーバ装置（サーバ）とともに分散処理システムを構築するクライアント装置（クライアント）は、サーバが提供するサービスを利用するのに必要なサービスモジュール（表示制御モジュール等）をサーバから取得するための初期設定モジュールをサーバから取得し、取得した初期設定モジュールの実行によって必要なサービスモジュールを取得し、取得したサービスモジュールを実行することでサービスの利用をおこなうサービス利用手段（「クライアントデーモン起動処理」と、「初期設定モジュール取得処理」および「初期設定モジュール処理」におけるクライアント側処理に相当）を備えたことを特徴とする。

【0063】この請求項19の発明によれば、クライアントは、サービス利用手段によって、サーバから初期設定モジュールを取得して実行するだけで、この初期設定モジュールの処理により、クライアントにおいて必要なサービスモジュールのみをサーバから取得することができるので、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続された場合であっても、サーバによって提供されるサービスがクライアント間において差異なく利用できるとともに、ユーザが、クライアントの種別を意識することなく、サーバの提供するサービスを容易に利用することができる。

【0064】また、請求項20において、請求項19に記載のクライアントは、初期設定モジュールが、クライアントの処理能力や負荷状態の調査をおこない（「初期設定モジュール処理」のステップS104cに相当）、その調査結果に応じて、サーバが提供するサービスを利用するのに必要なサービスモジュールのうちサーバから取得するサービスモジュールを選択する（「初期設定モジュール処理」のステップS105cに相当）ことを特徴とする。

【0065】この請求項20の発明によれば、クライアントにおいて実行されるサービス利用手段にクライアント

10

20

30

40

50

トの機能と処理能力を調査する調査機構が設けられ、初期設定モジュールの実行時にこの調査機構を実行して取得される調査結果に応じて、サーバから取得するサービスモジュールを選択することができるので、たとえば、調査結果がクライアントの処理能力の低下を示す場合に、実行可能なサービスモジュールだけを取得して実行し、サービス実行に必要な他のサービスモジュールの実行をサーバ等に依頼することが可能になり、クライアントにおいてストレスのないサービス利用が可能になる。

【0066】また、請求項21において、請求項19または20に記載のクライアントは、サーバから取得したサービスモジュールを管理するモジュール管理テーブル（クライアントデータ管理テーブル）を備え、サービス利用手段が、モジュール管理テーブルにサービスを利用するのに必要なサービスモジュールが登録されていない場合に、そのサービスモジュールをサーバから取得し、取得したサービスモジュールをモジュール管理テーブルに登録する（「初期設定モジュール処理」のステップS107cに相当）ことを特徴とする。

【0067】この請求項21によれば、クライアントデータ管理テーブルに、サービスを利用するのに必要なサービスモジュールが登録されていない場合に、そのサービスモジュールをサーバから取得して、クライアントデータ管理テーブルに登録するので、クライアントは、サービスの利用に必要な最低限のサービスモジュールだけを付属する記憶装置等に備えておくことができ、クライアントの資源の有効な活用を実現することができる。

【0068】また、請求項22において、請求項19～21のいずれか一つに記載のクライアントは、サービス利用手段が、サービスを利用するのに必要なサービスモジュールの一種としてサーバとクライアントとの双方の実行環境において実行可能な共通モジュールをサーバから取得し、クライアントの処理能力や負荷状態の情報を一定時間毎に取得するリソース監視機構（「システム監視スレッド処理」に相当）を設け、共通モジュールを実行している状態（「サービス起動処理」のステップS112cに相当）において、リソース監視機構によって取得された情報が処理能力の低下や負荷の増大等によるリソース不足状態を示す場合に、サーバに対してそのサーバ側での共通モジュールの起動を要求する（「クライアントにおけるモジュール実行環境移行処理」のステップS124cに相当）ことを特徴とする。

【0069】この請求項22によれば、クライアントが実行環境を問わない共通モジュールを実行している状態において、クライアントのリソース不足状態に対して、本来、このクライアントにおいて実行すべき共通モジュールを、サーバにおいて実行するように実行位置を移行することができるので、クライアントのリソース不足が生じた場合に、クライアントにおいて実行されていた共通モジュールを、リソースに余裕のあるサーバにおいて

実行できるようになり、クライアントのスループットの低下を回避することができる。

【0070】また、請求項23において、請求項22に記載のクライアントは、サービス利用手段が、複数の共通モジュールを実行している状態において、クライアントのリソース不足状態が解消されるまで、起動要求の対象となる共通モジュールの選択をおこない、前記サーバ装置に対して前記選択した共通モジュールの起動の要求を繰り返す（「クライアントにおけるモジュール実行環境移行処理」に相当）ことを特徴とする。

【0071】この請求項23によれば、実行されている複数の共通モジュールのうち、リソース不足が解消されるだけの共通モジュールを、サーバに対する起動要求の対象とするので、起動要求先のサーバのリソースへの大きな圧迫を回避することができる。

【0072】また、請求項24において、請求項22または23に記載のクライアントは、共通モジュールの起動をおこなうサーバの実行位置情報に基づいて共通モジュール実行位置テーブル（クライアントデータ管理テーブル）を備え、サービス利用手段が、モジュール実行位置テーブルに登録された実行位置情報に基づいて共通モジュールにアクセスする（「サービスモジュールアクセス処理」に相当）ことを特徴とする。

【0073】この請求項24によれば、モジュール実行位置テーブルによって共通モジュールの実行位置を管理するので、このモジュール実行位置テーブルを参照することで、所望の共通モジュールのアクセスが容易になる。

【0074】また、請求項25において、請求項19～24のいずれか一つに記載のクライアントは、サーバから更新されたサービスモジュールの更新の通知を受けた場合に、サーバから更新されたサービスモジュールを取得する更新モジュール取得手段（「サービスモジュール更新処理」のクライアント側処理に相当）を備えたことを特徴とする。

【0075】この請求項25によれば、更新モジュール取得手段によって、サーバからサービスモジュールの更新が通知された際に、更新されたサービスモジュールを取得するので、クライアントにおいて、ユーザ自らが、サーバにおけるモジュールの更新時点を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。

【0076】**【発明の実施の形態】**以下に、本発明にかかる分散処理システム、分散処理方法、サーバ装置およびクライアント装置の実施の形態を図面に基いて詳細に説明する。なお、この実施の形態によりこの発明が限定されるものではない。ここでは、まず、本発明にかかる分散処理システム、分散処理方法、サーバ装置およびクライアント装置において動作される各処理についての原理的な説明

をおこない、ついでそれら各処理によって動作する具体的な実施の形態を説明する。

【0077】本発明にかかる分散処理システムは、LANケーブル、電話回線もしくは専用回線等の通信回線網、すなわちネットワークを介して接続される一つ以上のサーバと一つ以上のクライアントとから構成され、クライアントが、主にサーバの提供するサービスをネットワーク上の共有資源として利用できるシステムである。特に、複数のサーバがネットワークに接続されることにより、サービス実行における負荷が分散され、クライアントの受けるサービスがストレスなく実行できる環境が提供される。

【0078】また、この分散処理システムは、通信インターフェースを共通にすることで、ハードウェア仕様やオペレーティングシステムと異なる複数種のクライアントが接続されてよい。

【0079】ここにおけるサービスは、クライアントの要求に対して、サーバがクライアントに種々のデータを提供することであり、データベースサービスや計算処理サービス等である。よって、サーバは、クライアントからの要求に従って目的とするサービスを実行するためのサーバ用のサービスプログラムを備え、クライアントもまた、サーバに対してサービスを要求するためのクライアント用のサービスプログラムを備える必要がある。

【0080】クライアントは、オペレーティングシステム、上記したクライアント用のサービスプログラム、その他のアプリケーションプログラムを実行するCPUと、ネットワークに接続するための通信インターフェースと、CPUによって実行されるプログラムやデータを記憶する記憶装置と、キーボードやマウス等の入力装置と、CRTや液晶ディスプレイ等の表示装置と、から構成され、通常のコンピュータシステムを用いることができる。また、サーバは、クライアントと同様な構成であるが、さらに、複数のCPUを備えてマルチプロセッサ対応の高速な処理を可能とすることや、大容量の記憶装置、各種サービスを提供するための周辺装置が接続される場合がある。なお、上記した記憶装置は、RAM、ROM等のメモリやハードディスク装置、さらにICカードメモリ、フロッピーディスク、光磁気ディスク、CD-ROM等の取り外し可能な記録媒体を装填可能とするドライブ装置を含む。

【0081】本発明にかかる分散処理システムでは、上記したサービスプログラムを個別に実行可能な複数のモジュールから構成し、サーバ用サービスプログラムとクライアント用サービスプログラムとはともに、モジュール単位で実行可能である。また、クライアント用サービスプログラムを構成するモジュール群は、サービスデータとしてサーバからクライアントへ配信することが可能であり、これらモジュール(以下、サービスモジュールと称する)の保守・管理は主にサーバによっておこなわれ

る。

【0082】図1は、提供されるサービス毎に用意されたサービスデータのモジュール構成を示す説明図である。図1において、サービスデータには、サーバ側でサービス実行をおこなうために必要なサーバ制御モジュール、クライアントに備えられた表示装置に対する表示制御をおこない、グラフィカルユーザインターフェース(GUI)および他の入出力制御をおこなう表示制御モジュール、クライアント種別毎に用意され、クライアントに必要なサービスモジュールのサーバからの取得方法を定義しかつクライアントの処理能力と負荷状態の調査をおこなう機能を有する初期設定モジュール、さらにサービスモジュールの実行位置をクライアント、サーバに依存しない共通モジュールに分割される。

【0083】上述したサーバ用サービスプログラムおよびクライアント用サービスプログラムは、図1に示すサービスデータにおいて、必要とするサービスモジュールを組合せることによって構成され、また各サービスモジュール(サーバ制御モジュール、表示制御モジュール、初期設定モジュール、共通モジュール)を複数備えることができる。

【0084】たとえば、サーバ用サービスプログラムは、いくつかのサーバ制御モジュールから構成され、クライアント用サービスプログラムは、一つの初期設定モジュールと、いくつかの表示制御モジュールと、いくつかの共通モジュールとから構成される。

【0085】図2は、サーバのシステム実行形態を示す説明図である。サーバは、オペレーティングシステムSOSの実行環境下において、複数のクライアントからの接続要求の受け付け、サービスモジュールの配送、サーバのサービス制御をおこなうサーバデーモンSD1が常駐している。この状態において、クライアントからのサービス実行要求が発生すると、サーバデーモンSD1中に、該当するサービスプログラムSSP1(Service-1)のサービスモジュールを実行するためのスレッドを生成し、このサービスモジュールを実行する。

【0086】また、サーバデーモンSD1には、サーバの処理能力と負荷状態を逐次管理するシステム監視スレッドSTH1が含まれており、一定時間間隔でサーバに対するシステム監視がおこなわれる。また、サーバは、記憶装置に、サーバおよびクライアントに登録されるサービスモジュールの情報を管理するサーバデータ管理テーブルSDT1を備えており、必要に応じてこのサーバデータ管理テーブルSDT1を参照し、サーバデーモンSD1内のスレッドとのデータの入出力をおこなう。

【0087】なお、サーバデーモンSD1は、オペレーティングシステムSOSに対してデータおよび命令の入出力をおこない、オペレーティングシステムSOSは、通信用のソケットAPI(Application Program Interface)を介してネットワ

ークとのデータの入出力、すなわちクライアントまたは他のサーバとの接続を確立している。

【0088】サーバデータ管理テーブルSDT1は、サーバの提供するサービスのサービス名と、このサービスを実行するのに必要なサービスモジュールのサービスモジュール名と、各サービスモジュールの版数および実行状態に関する情報を含むサービスモジュールデータと、から構成される。図3は、サーバデータ管理テーブルSDT1の各サービスモジュールに対するサービスモジュールデータを示した説明図である。

【0089】図3において、サーバデータ管理テーブルSDT1におけるサービスモジュールデータは、たとえば、対応する「サービスモジュール名」、対応するサービス名を示す「デフォルトサービス名」、サーバがサービスモジュールを配送したクライアントを示す「配送先クライアントアドレスリスト」、サービスモジュールの種類（サーバ制御モジュール、表示制御モジュール、初期設定モジュール、共通モジュール）を示す「モジュール種別」、クライアントのオペレーティングシステムまたはハードウェア仕様を示す「プラットフォーム種別」、サービスモジュールの「版数」、サービスモジュールの実行位置を示す「状態フラグ」、「状態フラグ」がローカル実行を示す場合のサービスモジュールに対するアクセスコードを示す「ローカル実行時のアクセスID」、「状態フラグ」がリモート実行を示す場合のサービスモジュールの属するサーバのアドレスを示す「リモート実行時のサーバアドレス」、「状態フラグ」がリモート実行を示す場合のサービスモジュールに対するアクセスコードを示す「アクセスID」等のデータから構成される。

【0090】図4は、クライアントのシステム実行形態を示す説明図である。クライアントは、オペレーティングシステムCOSの実行環境下において、サーバとの通信をおこなうクライアントデーモンCD1が常駐している。ここで、オペレーティングシステムCOSは、サーバのオペレーティングシステムSOSと必ずしも同じである必要はない。

【0091】この状態において、ユーザからのサービス実行要求が発生すると、クライアントデーモンCD1中に、該当するサービスプログラムCSP1 (Service 1) のサービスモジュールを実行するためのスレッドを生成し、このサービスモジュールを実行する。

【0092】また、クライアントデーモンCD1には、クライアントの処理能力と負荷状態を逐次管理するシステム監視スレッドCTH1が含まれており、一定時間間隔でクライアントに対するシステム監視がおこなわれる。

また、クライアントは、記憶装置に、サーバから取得してクライアントに登録されるサービスモジュールの情報を管理するクライアントデータ管理テーブルCDT1を備えており、必要に応じてこのクライアントデータ

管理テーブルCDT1を参照し、クライアントデーモンCD1内のスレッドとのデータの入出力をおこなう。

【0093】なお、クライアントデーモンCD1は、オペレーティングシステムCOSに対してデータおよび命令の入出力をおこない、オペレーティングシステムCOSは、サーバと同様に、通信用のソケットAPIを介してネットワークとのデータの入出力、すなわちサーバとの接続を確立している。

【0094】クライアントデータ管理テーブルCDT1は、クライアントに登録されたサービスのサービス名と、このサービスを実行するのに必要なサービスモジュールのサービスモジュール名と、各サービスモジュールの版数および実行状態に関する情報を含むサービスモジュールデータと、から構成される。図5は、クライアントデータ管理テーブルCDT1の各サービスモジュールに対するサービスモジュールデータを示した説明図である。

【0095】図5において、クライアントデータ管理テーブルCDT1におけるサービスモジュールデータは、たとえば、図3に説明した「サービスモジュール名」、「デフォルトサービス名」、「版数」、「状態フラグ」、「ローカル実行時のアクセスID」、「リモート実行時のサーバアドレス」、「アクセスID」と、サービスモジュールの取得先であるサーバのアドレスを示す「取得サーバアドレス」等のデータから構成される。

【0096】つぎに、図2および図4に示したサーバおよびクライアントの実行形態において、サービスが実行されるまでの流れをフローチャートにしたがって説明する。

【0097】(クライアントデーモンの起動処理) まず、クライアントとなる端末装置は、多くの場合、起動と同時に、オペレーティングシステムCOSの環境下において、クライアントデーモンCD1が実行される。これにより、クライアントデーモンCD1は、常にクライアントデーモンとして生成されたスレッドが実行されており、各スレッドの実行内容にしたがったデータの出入力がおこなわれる。

【0098】図6は、クライアントデーモンCD1の起動において実行される処理を示したフローチャートである。図6において、たとえば、オペレーティングシステムCOSの起動と同時に自動実行されるタスク起動プログラムによって、まず、クライアントデーモンCD1が起動される(ステップS61)。クライアントデーモンCD1はサーバからのメッセージを受信したり、クライアントで実行されるサービスモジュールからのメッセージ要求をサーバに送信する。

【0099】クライアントで実行されるサービスモジュールには、ユーザが入力装置を用いることでおこなわれるボタン押下やテキスト入力などのサービス選択やサービス実行に関するイベントを制御する表示制御モジュール

ル、クライアント種別に応じたサービスモジュールの取得を可能にする初期設定モジュール、クライアントがサーバからの実行位置に依存しない共通モジュールが含まれる。

【0100】このクライアントデーモンCD1の起動にともなう、クライアントの処理能力や負荷状態の監視をおこなうシステム監視スレッドCTH1が生成されて実行される(ステップS62)。このようにシステム監視スレッドCTH1が常駐した後は、現在サーバに登録されているサービスのリスト、すなわちサーバが提供可能なサービスのリストを取得する(ステップS63)。

【0101】実際には、クライアントデーモンCD1内には、サーバからのサービスリスト取得をおこなうためのスレッドが生成され、このスレッドがオペレーティングシステムCOSに対してサービスリストの取得を要求する。オペレーティングシステムCOS(より詳しくは通信ソケットAPI)は、クライアントがアクセス可能なサーバに対してサービスリストの配信を要求する。

【0102】このように、サービスリストの取得をおこなうスレッドもまた、サービスモジュールとして機能し、クライアントにおいて最低限必要なモジュールとなる。よって、ユーザは、クライアントデーモンCD1の起動時のみでなく、必要に応じてサーバからサービスリストの取得をおこなうことができる。

【0103】サーバからサービスリストが配信されると、上記したサービスリスト取得スレッドまたはあらかじめ登録された表示制御モジュールのスレッドが、クライアントの表示装置上にサービスリストにしたがったGUI表示をおこない、ユーザによる入力装置を介したアクセスもしくはサーバアクセスの待ち受け状態に移行する(ステップS64)。

【0104】ここで、サーバアクセスとは、サーバがクライアントに対しておこなうメッセージ送信やクライアント上で実行されているサービスモジュールに対する命令またはデータの送信を示す。

【0105】(サーバデーモンの起動処理)図6に示したステップS64の後の処理は後述するとして、つぎに、サーバデーモンSD1の起動処理について説明する。サーバとなる端末装置は、オペレーティングシステムSOSの環境下において、通常、サーバデーモンSD1が常駐状態であり、サーバデーモンとして生成されたスレッドが常に実行されている。サーバは、クライアントからのアクセスを常に受け付ける必要があるため、保守作業等の特別な場合を除いては、常に起動状態であると考えてよい。このため、サーバデーモンSD1の起動処理もまた稀にしかおこなわれないが、ここでは、サーバの起動時またはシステム管理者の操作によるサーバデーモンSD1の起動がおこなわれる場合を説明する。

【0106】図7は、サーバデーモンSD1の起動時に実行される処理を示したフローチャートである。図7に

おいて、まず、サーバデーモンSD1が起動される(ステップS71)。サーバデーモンSD1は、クライアントや他のサーバからのメッセージを受信したり、サーバ内で実行されるサービスモジュールからのメッセージ要求をクライアントや他のサーバに送信する。

【0107】サーバで実行されるサービスモジュールには、常駐型のサービスやクライアントからのサービス起動要求により起動されるサーバ制御モジュール、クライアントがサーバからの実行位置に依存しない共通モジュールが含まれる。

【0108】このサーバデーモンSD1の起動にともなう、サーバの処理能力や負荷状態の管理をおこなうシステム監視スレッドSTH1が生成されて実行される(ステップS72)。そして、システム監視スレッドSTH1が常駐した後は、クライアントからのアクセスもしくは他サーバからのアクセス待ち受け状態に移行する(ステップS73)。

【0109】(システム監視スレッドの処理)つぎに、上述した「クライアントデーモンCD1の起動処理」におけるシステム監視スレッドCTH1の実行(ステップS62)および「サーバデーモンの起動処理」におけるシステム監視スレッドSTH1の実行処理(ステップS72)について説明する。なお、システム監視スレッドCTH1およびSTH1は、ともに同一の流れの処理を示す。

【0110】図8は、システム監視スレッドによる処理を示したフローチャートである。システム監視スレッドは、クライアントおよびサーバにそれぞれ配備され、クライアントデーモンCD1およびサーバデーモンSD1によって生成される。つまり、システム監視スレッドは、このシステム監視スレッドが常駐しているシステム(サーバまたはクライアント)の処理能力や負荷状態のチェックをおこなう(ステップS81)。ここで、処理能力のチェックとしては、メモリや記憶装置のデータ蓄積容量、サーバやクライアントとの通信に必要な通信ソケットの空き数、スレッド生成の許容量等の調査があげられ、負荷状態のチェックとしては、CPU稼働率やベンチマークテストの実施による計測があげられる。

【0111】つづいて、ステップS81の処理能力や負荷状態のチェックの結果、システムが、処理能力の低下や著しい負荷状態、すなわちリソース不足の状態を示すかどうか判定される(ステップS82)。ステップS82において、システムのリソース不足の状態が示された場合には、さらに、このシステム監視スレッドの実行位置がクライアントであるかどうか判定される(ステップS84)。

【0112】ステップS84において、システム監視スレッドの実行位置がクライアントであれば、後述するクライアント用のモジュール実行環境移行処理(ステップS120)が実行され、システム監視スレッドの実行位

29

置がサーバであれば、後述するサーバ用のモジュール実行環境移行処理（ステップS140）が実行される。

【0113】一方、ステップS82において、リソース不足が検出されない場合には、一定時間の待ち受け状態（ステップS83）となり、この一定時間経過後に再びステップS81の処理がおこなわれる。すなわち、図8に示すシステム監視スレッドの処理は、一定時間毎にシステムの処理能力・負荷状態のチェックをおこない、そのチェックの結果がリソース不足を示す場合には、このシステム監視スレッドが実行されている位置（サーバまたはクライアント）に応じたモジュール実行環境移行処理が実行されることを示す。

【0114】（初期設定モジュール取得処理） つぎに、図6に示したステップS64のユーザアクセス/サーバアクセス待ち受け状態に対して、ユーザによるサービス選択がおこなわれ、その結果実行される初期設定モジュール取得処理について説明する。図9は、初期設定モジュール取得処理を示したフローチャートである。

【0115】図9において、まず、ユーザが、クライアントの表示装置上にGUI表示されたサービスリストから、マウスやキーボード等の入力装置を用いて、所望のサービスを選択する（ステップS91c）。そして、選択されたサービスを実行するのに必要な初期設定モジュールを、上記したクライアントデータ管理テーブルCDT1から検索する（ステップS92c）。

【0116】 ついて、該当する初期設定モジュールが、クライアントデータ管理テーブルCDT1に登録されているかどうかの判定がおこなわれる（ステップS93c）。ステップS93cにおいて、該当する初期設定モジュールがクライアントデータ管理テーブルCDT1に登録されている場合は、サーバに登録されている初期設定モジュールの版数をチェックするために、サーバに登録された初期設定モジュールのうち、上記した初期設定モジュールに対応する初期設定モジュールの情報を取得する（ステップS94c）。

【0117】ステップS94cの処理の手順としては、該当する初期設定モジュールを表すサービスモジュール名をとった初期設定モジュールの情報取得要求をサーバに対して発信する。そして、クライアントは、この初期設定モジュール情報取得要求に対してサーバから返信される返答、すなわち該当する初期設定モジュールの情報を受取る。

【0118】クライアントが受け取った初期設定モジュール情報は、たとえば図3に示すサービスモジュールデータのように構成されており、このデータに含まれる版数情報と、クライアントに登録された初期設定モジュール情報、たとえば図5に示すサービスモジュールデータに含まれる版数情報と、の比較をおこない、版数が一致するかどうかを判定する（ステップS95c）。

【0119】ステップS95cにおいて、版数が一致し

30

ない場合、すなわちクライアントに登録された初期設定モジュールの版数が、サーバに登録された初期設定モジュールの版数よりも古い値を示す場合には、クライアントにおいて該当する初期設定モジュールを変更する必要があるため、サーバに対して初期設定モジュールの取得要求を発信する。

【0120】また、ステップS93cにおいて、該当する初期設定モジュールがクライアントデータ管理テーブルCDT1に登録されていない場合もまた、サーバに対して初期設定モジュールの取得要求を発信する。なお、初期設定モジュール取得要求には、目的とする初期設定モジュールに対応したサービス名と、クライアントのハードウェア仕様またはオペレーティングシステムの種別を特定する数値情報が含まれている。

【0121】サーバは、クライアントから受信した初期設定モジュール取得要求の示す引数（上記した数値情報）によって、サービス名とクライアント種別を特定する（ステップS91s）。ついで、サーバは、サーバデータ管理テーブルSTD1から、ステップS91sにおいて特定されたサービス名とクライアント種別に対応した初期設定モジュールを選択する（ステップS92s）。

【0122】そして、サーバは、ステップS92sにおいて選択された初期設定モジュールを、初期設定モジュール取得要求のあったクライアントに対して配信する。サーバは、初期設定モジュールの配信後、サーバデータ管理テーブルSTD1において、配信した初期設定モジュールに対応するサービスモジュールデータのうち、「配送先クライアントアドレスリスト」に、初期設定モジュール取得要求のあったクライアントのアドレスを登録する（ステップS93s）。

【0123】一方、初期設定モジュールを受信したクライアント（より詳しくは、クライアントデーモンCD1）は、クライアントデータ管理テーブルCDT1に、受信した初期設定モジュールの登録をおこなう。この際、取得先のサーバのアドレスを、図5に示すサービスモジュールデータ内の「取得サーバアドレス」に登録する（ステップS96c）。

【0124】 ついて、クライアントは、ステップS96cにおいて登録された初期設定モジュールのスレッドを生成し（ステップS97c）、この初期設定モジュールを起動する（ステップS98c）。

【0125】なお、以上に説明した初期設定モジュール取得処理は、上述したサービスリスト取得スレッドまたはこのスレッドにより構成されるサービスプログラムに含めてもよいし、初期設定モジュール取得処理をおこなうためのサービスモジュールをクライアントにあらかじめ登録しておくことで実行されるようにしてもよい。

【0126】（初期設定モジュール処理） つぎに、図9

50

に示したステップS98cにつづく処理、すなわち初期設定モジュールの処理について説明する。図10は、初期設定モジュールによる処理を示したフローチャートである。図10において、まず、初期設定モジュールは、その内部に定義された情報から、サービスの実行に必要なサービスモジュールを抽出する（ステップS101c）。

【0127】そして、初期設定モジュールは、クライアントデータ管理テーブルCDT1から、ステップS101cにおいて抽出したサービスモジュールの検索をおこなう（ステップS102c）。ステップS102cにおいて、クライアントデータ管理テーブルCDT1から検索対象となるサービスモジュールを特定することができる場合、すなわち検索対象となるサービスモジュールがクライアントデータ管理テーブルCDT1に登録されている場合には、そのサービスモジュールの版数をチェックする（ステップS103c）。

【0128】ステップS103cにおけるサービスモジュールの版数のチェックは、まず、サーバに登録されているサービスモジュールの版数を取得するために、サーバに登録されたサービスモジュールのうち、対応するサービスモジュールの情報の取得要求をそのサーバに対して発信する必要がある。

【0129】クライアントは、このサービスモジュール情報取得要求の発信後、サーバから配信される返答、すなわち該当するサービスモジュール情報を受け取り、初期設定モジュールは、受け取ったサービスモジュール情報から版数情報を取得する。なお、このサービスモジュール情報取得要求をおこなうにおいて、複数のサービスモジュールの情報が必要である場合は、該当する複数のサービスモジュールの情報を一度の送受信によって取得することも可能である。

【0130】一方、クライアントに登録されたサービスモジュールに対しては、クライアントデータ管理テーブルCDT1において、対象となるサービスモジュールに対応するサービスモジュールデータから、版数情報を取得することができる。なお、ステップS102cにおいて、検索対象となるサービスモジュールがクライアントデータ管理テーブルCDT1に登録されていない場合には、ステップS103cの処理はスキップされ、つぎのステップS104cの処理が実行される。

【0131】ステップS104cにおいては、オペレーティングシステムCOSに対するシステムコールによって、現在のクライアントの機能、処理能力および負荷状態のチェックがおこなわれる。このクライアントに対するシステムチェックは、上述したシステム監視スレッドCTH1によっておこなうこともでき、その場合、初期設定モジュールは、システム監視スレッドCTH1に対してシステムがリソース不足状態にあるかどうかの情報の取得要求をおこなうことができるが、ここでは、

に、システムの迅速な状態チェックをおこなうために、システムコールを用いた初期設定モジュール独自のシステムチェックをおこなうものとする。

【0132】つづいて、初期設定モジュールは、ステップS101cにおいて抽出されたサービスモジュールをサーバから取得する必要があるかどうかの判定をおこなう（ステップS105c）。ステップS105cにおいて、サービスモジュールをサーバから取得する必要がある場合、具体的には、ステップS102cにおいてサービスモジュールがクライアントデータ管理テーブルCDT1に登録されていない場合や、ステップS103cにおいて取得した両者のサービスモジュールの版数情報が異なる場合には、サーバに対してサービスモジュールの取得要求を発信する。

【0133】サーバは、クライアントから受信したサービスモジュール取得要求に含まれる情報によって、サービスモジュールを特定し、サーバデータ管理テーブルSDT1から、特定されたサービスモジュールを選択する（ステップS101s）。

【0134】そして、サーバは、ステップS101sにおいて選択されたサービスモジュールを、サービスモジュール取得要求のあったクライアントに対して配信する。サーバは、サービスモジュールの配信後、サーバデータ管理テーブルSDT1において、配信したサービスモジュールに対応するサービスモジュールデータのうち、「配送先クライアントアドレスリスト」に、サービスモジュール取得要求のあったクライアントのアドレスを登録する（ステップS102s）。

【0135】一方、サービスモジュールを受信したクライアント（より詳しくは、クライアントデーモンCD1）は、まず、受信したサービスモジュールをクライアントデータ管理テーブルCDT1に登録するかどうかの判定をおこなう（ステップS106c）。ステップS106cの処理は、ステップS104cにおけるシステムチェック、またはシステム監視スレッドCTH1において、リソース不足状態が示された場合に、実行すべきサービスモジュール、特に共通モジュールの実行をサーバに依頼するための処理を示している。

【0136】ステップS106cにおいて、サービスモジュールの登録をおこなう場合、すなわちクライアントのリソース不足が検出されない場合には、クライアントデータ管理テーブルCDT1に、サーバから受信したサービスモジュールの登録をおこなう。この際、取得先のサーバのアドレスを、図5に示すサービスモジュールデータ内の「取得サーバアドレス」に登録する（ステップS107c）。

【0137】ステップS106cにおいて、サービスモジュールの登録をおこなわない場合、すなわちクライアントのリソース不足が検出された場合には、後述するクライアント用モジュール実行環境移行処理（図8のステ

33

ップS120)がおこなわれる。

【0138】そして、クライアント用モジュール実行環境移行処理がおこなわれた後、またはステップS107cの終了後に、以下に説明するサービスの起動処理がおこなわれる。

【0139】(サービス起動処理) つぎに、上述した図9のステップS91cにおいて選択されたサービスの起動処理について説明する。図11は、サービス起動処理を示したフローチャートである。図11において、まず、クライアントデーモンCD1は、ユーザからのサービス起動要求により、サーバに対してサーバ制御をおこなうサーバ制御モジュールの起動要求をおこなう。この要求によって、サーバは、該当するサーバ制御モジュールを起動する(ステップS111s)。

【0140】このとき、クライアント処理能力の低下が著しく、さらに過負荷状態の場合、すなわち、上述した「初期設定モジュール処理」のステップS106cにおいて、サービスモジュールの登録をおこなわない場合、またはシステム監視スレッドCTH1によって新たにリソース不足が検出された場合には、クライアントデーモンCD1は、本来自端末で実行すべきである共通モジュールの起動をサーバに依頼するために共通モジュール起動要求をおこなう。

【0141】よって、ステップS111sの処理につづく共通モジュール起動要求の有無の判定(ステップS112s)が必要となる。ステップS112sにおいて、共通モジュール起動要求が存在する場合には、まず、このサーバにおける処理能力・負荷状態のチェックがおこなわれる(ステップS113s)。すなわち、サーバデーモンSD1が、システム監視スレッドSTH1から、サーバのリソース状態を示す登録を取得する。

【0142】つづいて、ステップS113sによって取得された情報によって、システムがリソース不足であるかどうかの判定がおこなわれる(ステップS114s)。ステップS114sにおいてリソース不足が示された場合には、起動要求のあった共通モジュールの実行を、さらに他のサーバに依頼するためのサーバ用モジュール実行環境移行処理(ステップS140)が実行される。

【0143】ステップS114sにおいてリソース不足が検出されない場合には、サーバデーモンSD1は、起動を要求された共通モジュールの起動をおこない(ステップS115s)、その共通モジュールへのアクセスに必要な実行アドレス、アクセスID、状態フラグ(図3に示すサービスモジュールデータの「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」参照)を設定する(ステップS116s)。

【0144】そして、起動要求のあったクライアントに対して、サービスの起動、すなわちサーバ制御モジュールと共通モジュールの起動要求があった場合の共通モ

34

ジュールとの起動の完了を示し、かつステップS116sにおいて設定された実行アドレス、アクセスID、状態フラグの情報を含んだサービス起動完了通知を配信する。その後、サーバデーモンSD1は、クライアントアクセス待ち受け状態に遷移する(ステップS117s)。これは、サーバ用のサービスプログラム(サーバ制御モジュール、共通モジュール)の起動が完了した状態を示す。

【0145】クライアントデーモンCD1は、サーバからサービス起動完了通知を受信すると、実行するサービスに対応した表示制御モジュールを起動する(ステップS111c)。つづいて、実行すべき共通モジュールのうち、システム監視スレッドCTH1等によって取得されるクライアントのシステムリソース情報に応じて、実行できるだけの共通モジュールを起動する(ステップS112c)。

【0146】そして、クライアントデーモンCD1は、クライアントデータ管理テーブルCDT1に、サービス実行に必要な共通モジュールのサービスモジュールデータのそれぞれにおいて、「ローカル実行時のサーバアドレス」、「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」等の登録をおこなう(ステップS113c)。これにより、クライアント用のサービスプログラム(表示制御モジュール、共通モジュール)の起動が完了したことが示される。

【0147】よって、ユーザは、このクライアント用のサービスプログラムによってサーバへのアクセス(詳しくは、サーバ用のサービスプログラムへのアクセス)をおこなうことができ、サーバの提供するサービスを受けることができる。

【0148】(クライアントにおけるモジュール実行環境移行処理) つぎに、クライアントにおいて、システム監視スレッドCTH1によるシステムの処理能力の低下や過負荷状態が検出された場合に、共通モジュールの実行環境をサーバに移行する処理について説明する。すなわち、この処理は、図8に示すシステム監視スレッドの処理において、ステップS120のクライアント用モジュール実行環境移行処理に該当する。

【0149】図12は、クライアントにおけるモジュール実行環境移行処理を示したフローチャートである。なお、このモジュール実行環境移行処理は、クライアントにおいてあらかじめ登録されたモジュール実行環境移行プログラムにより提供され、そのスレッドは、クライアントデーモンCD1内に生成されることにより、システムに常駐して実行されるものとす。

【0150】図12において、まず、クライアントデーモンCD1は、クライアントデータ管理テーブルCDT1から現在実行中の共通モジュールのうち、一つ目の共通モジュールを抽出する(ステップS121c)。そして、クライアントデーモンCD1は、ステップS121

50

cにおいて、共通モジュールの抽出が可能であったかどうかを判定する（ステップS122c）。

【0151】ステップS122cにおいて共通モジュールの抽出が可能でなかった場合、すなわち実行に必要な共通モジュールがクライアントデータ管理テーブルCDT1にこれ以上登録されていなかった場合には、このモジュール実行環境移行処理を終了する。また、ステップS122cにおいて共通モジュールの抽出が可能であった場合、抽出された共通モジュールのサービスモジュールデータから状態フラグの情報を取得し、この状態フラグが「ローカル実行」を示すかどうかを判定する（ステップS123c）。

【0152】ステップS123cにおいて状態フラグが「ローカル実行」を示さない場合、すなわち抽出された共通モジュールがサーバにおいて実行されている場合は、実行環境を移行する必要がないので、クライアントデータ管理テーブルCDT1においてつぎの共通モジュールの検索をおこなうために、再びステップS121cの処理に戻る。

【0153】ステップS123cにおいて状態フラグが「ローカル実行」を示す場合、すなわち抽出された共通モジュールがクライアントにおいて実行されている場合は、サーバに対してこの共通モジュールの実行を依頼するための共通モジュール起動要求をおこなう（ステップS124c）。

【0154】サーバは、クライアントからの共通モジュール起動要求を受信すると、まず、このサーバにおける処理能力・負荷状態のチェックをおこなう（ステップS121s）。すなわち、サーバデーモンSD1が、システム監視スレッドSTH1から、サーバのリソース状態

を示す情報を取得する。

【0155】つづいて、ステップS121sにおいて取得された情報によって、システムがリソース不足であるかどうかの判定がおこなわれる（ステップS122s）。ステップS122sにおいてリソース不足が検出された場合には、起動要求のあった共通モジュールの実行を、さらに他のサーバに依頼するためのサーバ用モジュール実行環境移行処理（ステップS140）が実行される。

【0156】ステップS122sにおいてリソース不足が検出されない場合には、サーバデーモンSD1は、起動要求された共通モジュールの起動をおこない（ステップS123s）、その共通モジュールへのアクセスに必要な実行アドレス、アクセスID、状態フラグ（図3に示すサービスモジュールデータの「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」参照）を設定する（ステップS124s）。

【0157】そして、起動要求のあったクライアントに対して、共通モジュールの起動の完了を示しかつステップS124sにおいて設定された実行アドレス、アクセ

スID、状態フラグの情報を含んだ共通モジュール起動完了通知を配信する。

【0158】クライアントデーモンCD1は、サーバから共通モジュール起動完了通知を受信すると、ローカル実行中の共通モジュールの実行を停止し、蓄積領域が不足している場合は共通モジュールプログラムを削除する。つづいて、クライアントデーモンCD1は、クライアントデータ管理テーブルCDT1において、対象となる共通モジュールのサービスモジュールデータの状態フラグを「リモート実行」に変更する（ステップS125c）。

【0159】そして、クライアントデーモンCD1は、クライアントデータ管理テーブルCDT1に、サービス実行に必要な共通モジュールのサービスモジュールデータのそれぞれにおいて、「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」の登録をおこなう（ステップS126c）。

【0160】さらに、クライアントデーモンCD1は、システム監視スレッドCTH1からシステムリソース情報を取得することにより、システムがリソース不足（処理能力の低下または過負荷状態）であるかどうかを判定する（ステップS127c）。ステップS127cにおいてリソース不足が検出されない場合は、このモジュール実行環境移行処理を終了する。

【0161】ステップS127cにおいてリソース不足が検出された場合は、再びステップS121cの処理を繰り返す。すなわち、このモジュール実行環境移行処理は、システムのリソース不足が解消されるまで繰り返される。また、本発明にかかる分散処理システムでは、共通モジュールは他のサービスモジュールやクライアントデーモンCD1によって共有されるため、共通モジュールの実行位置が変更となった場合であっても同様に共通モジュールへのアクセスを保証する必要がある。

【0162】図13は、サービスモジュールアクセス処理を示したフローチャートである。クライアントデーモンCD1または他のサービスモジュールは、実行に必要なサービスモジュールへのアクセスに先だって、図13に示すサービスモジュールアクセス処理をおこなう。図13において、まず、クライアントデーモンCD1または他のサービスモジュールは、クライアントデータ管理テーブルCDT1から、該当するサービスモジュール（ここでは、特に共通モジュール）のサービスモジュールデータを検索する（ステップS131c）。なお、このサービスモジュールアクセス処理を実行するまでに、上述した初期設定モジュールによるサービスモジュールのセットアップは完了しているものとする。

【0163】つづいて、クライアントデーモンCD1または他のサービスモジュールは、ステップS131cにおいて検索されたサービスモジュールデータから状態フラグの情報を取得し、状態フラグが「ローカル実行」を

37

示すかを判定する(ステップS132c)。ステップS132cにおいて状態フラグが「ローカル実行」でない場合、すなわち状態フラグが「リモート実行」を示す場合には、サービスモジュールデータに含まれた「リモート実行時のサーバアドレス」の示すサーバに対し、このサービスモジュールに対する処理を要求する。なお、この要求の通知には、該当するサービスモジュールにアクセスするための「アクセスID」の情報も含まれている。

【0164】サーバデーモンSD1は、クライアントからのサービスモジュール処理要求を受信して、この要求から処理対象となる「アクセスID」を特定し、この「アクセスID」を用いることで、該当するサービスモジュールのアクセスを達成する(ステップS131s)。

【0165】そして、ステップS132cにおいて状態フラグが「ローカル実行」を示す場合には、該当するサービスモジュールのサービスモジュールデータから「ローカル実行時のアクセスID」を取得し、このアクセスIDを用いることで、該当するサービスモジュールのアクセスを達成する(ステップS133c)。

【0166】よって、このサービスモジュールアクセス処理によれば、共通モジュールの実行位置がクライアントであってもサーバであっても他のサービスモジュールやクライアントデーモンCD1に影響を受けることはない。また、共通モジュールの実行位置は「リモート実行時のサーバアドレス」と「アクセスID」によって規定されるため、リモート実行をおこなうサーバはクライアントが共通モジュールを取得したサーバに限定されるものではない。

【0167】(サーバにおけるモジュール実行環境移行処理) つぎに、サーバにおいて、システム監視スレッドSTH1によってシステムの処理能力の低下や過負荷状態が検出された場合に、共通モジュールの実行環境を他のサーバに移行する処理について説明する。すなわち、この処理は、図8に示すシステム監視スレッドの処理において、ステップS140のサーバ用モジュール実行環境移行処理に該当する。

【0168】図14は、サーバにおけるモジュール実行環境移行処理を示したフローチャートである。なお、このモジュール実行環境移行処理は、サーバにおいてあらかじめ登録されたモジュール実行環境移行プログラムにより提供され、そのスレッドは、サーバデーモンSD1内に生成されることにより、システムに常駐して実行されるものとする。

【0169】図14において、まず、サーバデーモンSD1は、このサーバに対してサービスモジュールの起動要求が発生しているかどうかを判定する(ステップS141s1)。ステップS141s1においてサービスモジュールの起動要求が発生していない場合は、サーバデー

38

ータ管理テーブルSDT1から現在実行中のサービスモジュールのうち、一つ目のサービスモジュールを抽出する(ステップS142s1)。そして、サーバデーモンSD1は、ステップS142s1において、サービスモジュールの抽出が可能であったかどうかを判定する(ステップS143s1)。

【0170】ステップS143s1においてサービスモジュールの抽出が可能でなかった場合には、このモジュール実行環境移行処理を終了する。また、ステップS143s1においてサービスモジュールの抽出が可能であった場合、抽出されたサービスモジュールのサービスモジュールデータから状態フラグの情報を取得し、この状態フラグが「ローカル実行」を示すかどうかを判定する(ステップS144s1)。

【0171】ステップS144s1において状態フラグが「ローカル実行」を示さない場合、すなわち抽出されたサービスモジュールが他のサーバにおいて実行されている場合は、実行環境を移行する必要がないので、サーバデータ管理テーブルSDT1においてつぎのサービスモジュールの検索をおこなうために、再びステップS142s1の処理に戻る。

【0172】ステップS144s1において状態フラグが「ローカル実行」を示す場合、すなわち抽出されたサービスモジュールがこのサーバ(第1のサーバ)において実行されている場合は、他のサーバ(第2のサーバ)に対してこのサービスモジュールの実行を依頼するためのサービスモジュール起動要求をおこなう(ステップS145s1)。

【0173】他のサーバ(第2のサーバ)は、上記したサーバからサービスモジュール起動要求を受信すると、まず、このサーバにおける処理能力・負荷状態のチェックをおこなう(ステップS141s2)。すなわち、第2のサーバのサーバデーモンが、第2のサーバのシステム監視スレッドから、第2のサーバのリソース状態を示す情報を取得する。

【0174】つづいて、ステップS141s2において取得された情報によって、システムがリソース不足であるかどうかの判定がおこなわれる(ステップS142s2)。ステップS142s2においてリソース不足が検出された場合には、起動要求のあったサービスモジュールの実行を、さらに他のサーバ(第3のサーバ)に依頼するためのサーバ用モジュール実行環境移行処理、すなわちこのモジュール実行環境移行処理が、第3のサーバに対して再帰的に実行される。

【0175】ステップS142s2においてリソース不足が検出されない場合には、第2のサーバのサーバデーモンは、起動要求されたサービスモジュールの起動をおこなう(ステップS143s2)、そのサービスモジュ

ールへのアクセスに必要な実行アドレス、アクセスID、状態フラグ(図3に示すサービスモジュールデータの「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」参照)を設定する(ステップS144s2)。

【0176】そして、起動要求を発した第1のサーバに対して、サービスモジュールの起動の完了を示しかつステップS144s2において設定された実行アドレス、アクセスID、状態フラグの情報を含んだサービスモジュール起動完了通知を配信する。

【0177】第1のサーバのサーバデーモンSD1は、第2のサーバまたは他のサーバからサービスモジュール起動完了通知を受信すると、ローカル実行中のサービスモジュールの実行を停止する。つまり、第1のサーバのサーバデーモンSD1は、サーバデータ管理テーブルSDT1において、対象となるサービスモジュールのサービスモジュールデータの状態フラグを「リモート実行」に変更する(ステップS146s1)。

【0178】そして、第1のサーバのサーバデーモンSD1は、サーバデータ管理テーブルSDT1に、サービス実行に必要なサービスモジュールのサービスモジュールデータのそれぞれにおいて、「リモート実行時のサーバアドレス」、「アクセスID」、「状態フラグ」の登録をおこなう(ステップS147s1)。

【0179】さらに、第1のサーバのサーバデーモンSD1は、システム監視スレッドSTH1からシステムリソース情報を取得することにより、システムリソース不足(処理能力の低下または過負荷状態)であるかどうかを判定する(ステップS148s1)。ステップS148s1においてリソース不足が検出されない場合は、このモジュール実行環境移行処理を終了する。

【0180】ステップS148s1においてリソース不足が検出された場合は、再びステップS142s1の処理を繰り返す。すなわち、このモジュール実行環境移行処理は、システムのリソース不足が解消されるまで繰り返される。以上に説明したサーバにおけるモジュール実行環境移行処理によってサービスモジュールの実行位置が変更となった場合であっても、図13に示すサービスモジュールアクセス処理によってサービスモジュールへのアクセスを保証することができる。

【0181】(サービスモジュール更新処理) つぎに、サーバに登録されているサービスモジュールが更新された場合に、サーバデーモンSD1が自動的にクライアントに対してサービスモジュールの更新登録を要求する処理について説明する。

【0182】図15は、サービスモジュール更新処理を示したフローチャートである。図15において、まず、システム管理者等によって、サーバに登録されたサービスモジュールの更新がおこなわれる(ステップS151s)。

そして、システム管理者等による指示またはサー

バデーモンSD1によって、サーバデータ管理テーブルSDT1において、更新対象となったサービスモジュールのサービスモジュールデータから、配送先クライアントアドレスリストを取得する(ステップS152s)。

【0183】つまりサーバデーモンSD1は、ステップS152sにおいて取得した配送先クライアントアドレスリストに示されるクライアントに対して、サービスモジュールの更新の通知となるサービスモジュール更新要求を配信する(ステップS153s)。このサービスモジュール更新要求には、配信対象となるクライアントに応じた初期設定モジュールの版数情報が含まれている。

【0184】ここで、サーバデーモンSD1は、配信対象となるクライアントがネットワークにログインしているかどうかのチェックをおこない(ステップS154s)、ネットワーク障害や端末未起動によって、クライアントのログイン状態が確認できない場合は、そのクライアントに対するサービスモジュール更新要求の配信をエラー終了としてスキップし、配信対象となる他のクライアントに対してのログインチェックを繰り返す。なお、サーバデーモンSD1は、このエラー終了によってサービスモジュール更新要求が配信されなかったクライアントに対して、ある一定時間毎にログインチェックをおこない、サービスモジュール更新要求の配信をこころみる。

【0185】一方、サービスモジュール更新要求を受信したクライアントデーモンCD1は、クライアント管理データテーブルCDT1において初期設定モジュールのサービスモジュールデータから、版数情報を取得し、サービスモジュール更新要求に含まれる初期設定モジュールの版数情報との比較をおこない、初期設定モジュールの変更をおこなうべきかを判定する(ステップS151c)。

【0186】ステップS151cにおいて、初期設定モジュールの変更をおこなう必要がある場合、すなわち、クライアント管理データテーブルCDT1に登録された初期設定モジュールの版数よりサービスモジュール更新要求に含まれる初期設定モジュールの版数が新しい場合は、サーバに対して初期設定モジュールの取得要求を発信する。なお、初期設定モジュール取得要求には、上述した「初期設定モジュール取得処理」と同様に、目的とする初期設定モジュールに対応したサービス名と、クライアントのハードウェア仕様またはオペレーティングシステムの種類を特定するプラットフォーム種別(クライアント種別)とを特定する数値情報が含まれている。

【0187】サーバは、クライアントから受信した初期設定モジュール取得要求の示す数値情報によって、サービス名とクライアント種別を特定し、サーバデータ管理テーブルSTD1から、特定されたサービス名とクライアント種別に対応した初期設定モジュールを選択する。

41

つづいて、サーバは、この選択された初期設定モジュールを、初期設定モジュール取得要求のあったクライアントに対して配信する。また、サーバは、初期設定モジュールの配信後、サーバデータ管理テーブルSDT1において、配信した初期設定モジュールに対応するサービスモジュールデータのうち、「配送先クライアントアドレスリスト」に、初期設定モジュール取得要求のあったクライアントのアドレスを登録する。

【0188】クライアントデーモンCD1は、サーバから初期設定モジュールを取得後、この初期設定モジュールをクライアントデータ管理テーブルCDT1に登録する。そして、クライアントデーモンCD1は、上記したクライアントデータ管理テーブルCDT1への初期設定モジュールの登録後、またはステップS151cにおいて初期設定モジュールの変更を必要としない場合に、該当する初期設定モジュールを実行する（ステップS152c）。

【0189】つづいて、クライアントデーモンCD1は、初期設定モジュールに定義されたサービスモジュールをサーバから取得する必要があるかどうかの判定をおこなう（ステップS153c）。ステップS153cにおいて、サービスモジュールをサーバから取得する必要がある場合、具体的には、該当するサービスモジュールがクライアントデータ管理テーブルCDT1に登録されていない場合や、登録されたサービスモジュールの版数情報が古い場合に、サーバに対してサービスモジュールの取得要求を発信する。

【0190】サーバは、クライアントから受信したサービスモジュール取得要求に含まれる情報によって、サービスモジュールを特定し、サーバデータ管理テーブルSDT1から、特定されたサービスモジュールを選択する。そして、サーバは、選択されたサービスモジュールを、サービスモジュール取得要求のあったクライアントに対して配信する。また、サーバは、サービスモジュールの配信後、サーバデータ管理テーブルSDT1において、配信したサービスモジュールに対応するサービスモジュールデータのうち、「配送先クライアントアドレスリスト」に、サービスモジュール取得要求のあったクライアントのアドレスを登録する。

【0191】一方、サービスモジュールを受信したクライアントは、クライアントデータ管理テーブルCDT1に、サーバから受信したサービスモジュールの登録をおこなう（ステップS154c）。この際、取得先のサーバのアドレスを、図5に示すサービスモジュールデータ内の「取得サーバアドレス」に登録する。よって、ユーザは、このサービスモジュール更新処理により、サービスモジュールの更新を意識することなしに常に最新のサービスモジュールの版数によるサービスを楽しむことが可能になる。

【0192】なお、以上に説明したサービスモジュール

42

更新処理は、たとえば、サービスモジュール更新処理をおこなうためのサービスモジュールをクライアントにあらかじめ登録しておくことで実行される。

【0193】つぎに、以上に説明した処理に基づいて動作する本発明の分散処理システムについて、より理解を深めるために、さらに具体的な実施の形態について説明する。

【0194】（実施の形態1）図16は、実施の形態1にかかわる分散処理システムのサービス未起動時のシステム構成を示した説明図である。図16において、実施の形態1にかかわる分散処理システムは、サーバとしてサービス提供をおこなうServer#1とクライアントとしてサービスを受けるClient#1が、ネットワークを介し、かつ通信インターフェースを共通にして接続されることで構成される。

【0195】なお、実際には、本発明にかかわる分散処理システムは、多数のサーバとクライアントがネットワークに接続されて構成されるが、ここでは、理解を容易にするため、一つのサーバと一つのクライアント間における処理を中心に説明する。

【0196】ここで、Server#1のネットワーク上の論理アドレスを#0001とし、Client#1のネットワーク上の論理アドレスを#0011とする。Server#1は、クライアントからの接続要求を受け付け、サービスの実行制御をおこなうサーバデーモンSD1と、サービスデータの管理をおこなうサーバデータ管理テーブルSDT1と、を備えている。また、Server#1は、図7に示したサーバデーモン起動処理がおこなわれた後の状態であり、クライアントアクセス／他のサーバアクセスの待ち受け状態にある。よって、サーバデーモンSD1内には、クライアントの処理能力や負荷状態の管理をおこなうシステム監視スレッドSTH1が生成されている。

【0197】Client#1は、Server#1との通信をおこない、ユーザにインタフェースを提供するクライアントデーモンCD1と、サーバから取得したサービスデータの管理をおこなうクライアントデータ管理テーブルCDT1と、を備えている。また、Client#1は、図6に示したクライアントデーモン起動処理がおこなわれた後の状態であり、ユーザアクセス／サーバアクセスの待ち受け状態にある。よって、クライアントデーモンCD1内には、クライアントの処理能力や負荷状態の管理をおこなうシステム監視スレッドCTH1が生成されている。

【0198】また、Server#1とClient#1の実行環境下において動作され、通信ソケットAPIを介してネットワークとのデータ入出力をおこなう。ここで、上述したように、Server#1とClient#1は、通信インターフェースを共通にしているため、必ずしも通

50

43

ペレーティングシステムOSが同一である必要はない。

【0199】ここで、サーバによって提供されるサービスの一つとしてService #1が実行される場合を考える。図17は、Service #1を実行するために必要なサービスデータを示す説明図である。すなわち、サーバ用サービスプログラムおよびクライアント用サービスプログラムは、このサービスデータに含まれるサービスモジュールの組合せによって構成される。

【0200】図17において、サービスデータService #1は、複数のサービスモジュールから構成され、具体的には、サーバで実行されサービス制御をおこなうサーバ制御モジュールServiceModule #A1~An、クライアントで実行されGUI表示やサーバとの通信をおこなう表示制御モジュールServiceModule #B1~Bn、クライアントで実行されサービス実行に必要なモジュールの取得方法やシステムの処理能力および負荷状態の調査が定義された初期設定モジュールServiceModule #C1~Cn、実行環境をサーバかクライアントかに依存しない共通モジュールServiceModule #D1~Dn 20

【0201】まず、図16に示すようなServer #1とClient #1のサービス未起動状態において、ユーザが、図6に示したクライアントデモン起動処理によってClient #1の表示装置上に提示されたサービスリストから、所望のサービスを選択する。このサービス選択により、図9に示した初期設定モジュール取得処理が実行され、Server #1に対する初期設定 30

モジュールの取得要求がおこなわれる。なお、ここでは、Client #1のクライアントデータ管理テーブルCDT1に、該当するサービスの実行に必要な初期設定モジュール、表示制御モジュールおよび共通モジュールが登録されていないものとする。

【0202】そして、Server #1は、Client #1のクライアント種別に応じた初期設定モジュールServiceModule #C1を、Client #1に返送する。一方、Client #1は、Server #1から初期設定モジュールServiceModule #C1を受け取り、初期設定モジュールServiceModule #C1の起動をおこなう。この初期設定モジュールServiceModule #C1の起動によって、図10に示した初期設定モジュール処理が実行され、初期設定モジュールServiceModule #C1内の定義内容から、Service #1の実行に必要な表示制御モジュールServiceModule #B1および共通モジュールServiceModule #D1の取得がおこなわれる。

【0203】ここで、Client #1において、Se 50

44

rvice #1を実行するのには、さらに共通モジュールServiceModule #D2が必要であったが、システム監視スレッドCTH1によりClient #1のリソース不足が検出されたために、共通モジュールServiceModule #D2の取得および起動はおこなわれず、共通モジュールServiceModule #D2の起動をServer #1に依頼するものとする。

【0204】Client #1は、Server #1に対してService #1の起動要求をおこない、図11に示したサービス起動処理が実行される。この際、Server #1は、Client #1から起動要求のあった共通モジュールServiceModule #D2を起動して、クライアントアクセス待ち受け状態に移す。

【0205】つづいて、Client #1において、表示制御モジュールServiceModule #B1、共通モジュールServiceModule #D1の起動がおこなわれ、Service #1の起動が完了する。

【0206】図18は、実施の形態1にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。図18において、Server #1は、図16に示す状態に加えて、Service #1を提供するサーバ用サービスプログラムとして、サーバ制御モジュールServiceModule #A1、共通モジュールServiceModule #D2が用意され、これらサービスモジュールの実行においてサーバデモンSD1内にスレッドが生成される。

【0207】また、Client #1は、図16に示す状態に加えて、Service #1を提供するクライアント用サービスプログラムとして、表示制御モジュールServiceModule #B1、初期設定モジュールServiceModule #C1、共通モジュールServiceModule #D1が用意され、これらサービスモジュールの実行においてクライアントデモンCD1内にスレッドが生成される。

【0208】つぎに、図9に示した初期設定モジュール取得処理において、Server #1がクライアントに初期設定モジュールの配信をおこなった場合のServer #1のサービスモジュールデータについて説明する。ここでは特に、Server #1が複数のクライアント(Client #1、Client #2、Client #3)に対して、初期設定モジュールServiceModule #C1を配信した場合を考える。

【0209】図19は、この場合のServer #1のサーバデータ管理テーブルSDT1に登録された初期設定モジュールServiceModule #C1のサービスモジュールデータを示す説明図である。図19において、まず、「サービスモジュール名」として「Ser

45

viceModule #C1」が登録され、「デフォルトサービス名」として「Service #1」が登録されている。そして、「配送先クライアントリスト」には、上記した複数のクライアントClient #1、Client #2、Client #3のネットワーク上における論理アドレス（#0011、#0012、#0013）が登録されている。

【0210】また、「モジュール種別」として初期設定モジュールを示すコードである「3」が登録され、「プラットフォーム種別」として、Client #1のハードウェア仕様またはオペレーティングシステムを特定するコード「1」が登録されている。

【0211】さらに、「版数」として「1」が登録されており、初期設定モジュールServiceModule #C1が更新された場合には、この「版数」が更新されて、図15に示したサービスモジュール更新処理がおこなわれる。ここでは、初期設定モジュールServiceModule #C1に対するサービスモジュール更新処理は、「配送先クライアントリスト」に登録されているClient #1、Client #2、Client #3に対して、ユーザが介することのない自動的な処理としておこなわれる。

【0212】つぎに、Client #1のクライアントデータ管理テーブルCDT1に登録されるサービスモジュールデータの例として、共通モジュールServiceModule #D1のサービスモジュールデータについて説明する。

【0213】図20は、この場合のClient #1のクライアントデータ管理テーブルCDT1に登録された共通モジュールServiceModule #D1のサービスモジュールデータを示す説明図である。図20において、まず、「サービスモジュール名」として「ServiceModule #D1」が登録され、「デフォルトサービス名」として「Service #1」が登録されている。そして、「版数」として「1」が登録され、「取得サーバアドレス」としてこの共通モジュールServiceModule #D1を配信したサーバの論理アドレスを示す「#0001」が登録されている。

【0214】さらに、「状態フラグ」としてローカル実行、すなわちClient #1での共通モジュールServiceModule #D1の実行を示すコードである「1」が登録され、「ローカル実行時のアクセスID」としてこの共通モジュールServiceModule #D1にアクセスするために必要なアクセスID「111」が登録されている。

【0215】つぎに、図8に示したシステム監視スレッド処理により処理能力の低下もしくは過負荷状態が検出され、図13に示したサービスモジュールアクセス処理がおこなわれる場合のServer #1およびClient #1のサービスモジュールデータについて説明す

46

る。ここでは特に、図18に示した状態から、Client #1において過負荷状態が検出され、Client #1がServer #1に対して共通モジュールServiceModule #D1の実行依頼をおこなった場合を考える。

【0216】図21は、実施の形態1にかかる分散処理システムのサービス起動後のシステム構成を示した説明図であり、特に図18に示した状態において、Server #1がClient #1から依頼された共通モジュールServiceModule #D1を起動した状態を示す。よって、図21に示すように、Client #1は、共通モジュールServiceModule #D1をクライアント用のサービスプログラムから開放し、Server #1が共通モジュールServiceModule #D1を起動する。

【0217】Server #1において共通モジュールServiceModule #D1が起動されると、サーバデータ管理テーブルSDT1の共通モジュールServiceModule #D1のサービスモジュールデータが変更される。図22は、この場合のServer #1のサーバデータ管理テーブルSDT1に登録された共通モジュールServiceModule #D1のサービスモジュールデータを示す説明図である。

【0218】図22において、まず、「サービスモジュール名」として「ServiceModule #D1」が登録され、「デフォルトサービス名」として「Service #1」が登録されている。そして、「配送先クライアントリスト」には、上記したクライアントClient #1の論理アドレスである「#0011」が登録されている。

【0219】また、「モジュール種別」として共通モジュールを示すコードである「4」が登録され、「プラットフォーム種別」として、サーバ、クライアント、ハードウェア仕様またはオペレーティングシステムを特定しないコード「0」が登録されている。

【0220】さらに、「版数」として「1」が登録され、「状態フラグ」としてローカル実行、すなわちServer #1での共通モジュールServiceModule #D1の実行を示すコードである「1」が登録されている。そして、「ローカル実行時のアクセスID」としてこの共通モジュールServiceModule #D1にアクセスするために必要なID「222」が登録されている。

【0221】一方、Client #1においてもまた、Server #1におけるモジュール実行が完了すると、クライアントデータ管理テーブルCDT1内の共通モジュールServiceModule #D1のサービスモジュールデータが変更される。図23は、この場合のClient #1のクライアントデータ管理テーブルCDT1に登録された共通モジュールServiceM

50

47

odule #D1のサービスモジュールデータを示す説明図である。

【0222】図23において、まず、「サービスモジュール名」として「ServiceModule #D1」が登録され、「デフォルトサービス名」として「Service #1」が登録されている。そして、「版数」として「1」が登録され、「取得サーバアドレス」としてこの共通モジュールServiceModule #D1を配信したサーバの論理アドレスを示す「#0001」が登録されている。

【0223】さらに、「状態フラグ」としてリモート実行、すなわちサーバによる共通モジュールServiceModule #D1の実行を示すコードである「2」が登録されている。そして、「リモート実行時のサーバアドレス」としてこの共通モジュールServiceModule #D1の起動をおこなうサーバであるServer #1のアドレスを示す「#0001」が登録され、「アクセスID」としてこのServer #1上の共通モジュールServiceModule #D1にアクセスするのに必要なID「222」が登録されている。

【0224】以上に説明したように、実施の形態1にかかる分散処理システムによれば、上述した「クライアントデーモンの起動処理」、「サーバデーモンの起動処理」、「初期設定モジュール取得処理」、「初期設定モジュール処理」、「サービス起動処理」の実行によって、ハードウェア仕様やオペレーティングシステムの異なるクライアントがネットワークに接続された場合であっても、クライアントに応じた初期設定モジュールを取得するだけで、サーバによって提供されるサービスがクライアント間において差異なく利用でき、また、クライアントのサービス利用手段によって、該当する初期設定モジュールを自動的に取得することができるので、サービスに必要なモジュール構成の知識をユーザに要求することなく、サーバの提供するサービスを容易に利用することが可能となる。

【0225】また、上述した「システム監視スレッドの処理」、「クライアントにおけるモジュール実行環境移行処理」の実行によって、クライアントの処理能力の低下が示される場合に、クライアントにおいて実行されていた共通モジュールの一部をサーバ上で起動することができ、クライアントのリソース不足によるスループットの低下を迅速に回避することができるとともに、効果的な負荷分散を実現することができる。

【0226】さらに、上述した「サービスモジュール更新処理」の実行によって、クライアントを利用するユーザは特にサーバにおけるモジュールの更新時期を意識する必要がなくなり、クライアントにおいて迅速なモジュールの更新処理が可能となる。

【0227】（実施の形態2）つぎに、実施の形態1に

48

において説明した図19および図20のサービスモジュールデータの登録内容に対応する状態、すなわちClient #1がServer #1に対して、共通モジュールServiceModule #D2の起動要求をおこなった状態（図18に示した状態）において、Server #1のリソース不足により、さらに、他のサーバであるServer #2に対して共通モジュールServiceModule #D2の起動要求がおこなわれる場合について説明する。

【0228】図24は、実施の形態2にかかる分散処理システムのサービス起動時のシステム構成を示した説明図である。図24において、実施の形態2にかかる分散処理システムは、サーバとしてService #1の提供をおこなうことのできるServer #1およびServer #2と、クライアントとしてService #1を受けるClient #1とが、ネットワークを介し、かつ通信インターフェースを共通にして接続されることで構成されている。

【0229】なお、実際には、本発明にかかる分散処理システムは、多数のサーバとクライアントがネットワークに接続されて構成されるが、ここでは、理解を容易にするため、二つのサーバと一つのクライアント間における処理を中心に説明する。

【0230】ここで、Server #1、Server #2のネットワーク上の論理アドレスをそれぞれ#0001、#0002とし、Client #1のネットワーク上の論理アドレスを#0011とする。Server #1およびServer #2はともに、図11に示したサービス起動処理を実行可能とする状態にあり、特に、Server #1は、図8に示したシステム監視スレッド処理によるリソース不足の検出によって、図14に示したモジュール実行環境移行処理が実行された状態にある。

【0231】ここでは、図14に示したモジュール実行環境移行処理における他のサーバが、Server #2に該当し、このServer #2によって、Client #1から起動要求のあった共通モジュールServiceModule #D2の起動がおこなわれる。よって、図24に示すように、Server #2においては、Service #1を提供するサーバ用サーバプログラムとして、共通モジュールServiceModule #D2が用意され、このサービスモジュールの実行においてサーバデーモンSD2内にスレッドが生成された状態にある。

【0232】一方、Client #1は、実施の形態1において説明した図18に示す状態と同様に、Server #1を提供するクライアント用サーバプログラムとして、表示制御モジュールServiceModule #B1、初期設定モジュールServiceModule #C1、共通モジュールServiceModu

50

49

le #D1が用意され、これらサービスモジュールの実行においてクライアントデーモンCD1内にスレッドが生成された状態にある。

【0233】また、Server #1、Server #2、Client #1は、オペレーティングシステムOSの実行環境下において動作され、通信ソケットAPIを介してネットワークとのデータ入出力をおこなう。ここで、上述したように、Server #1、Server #2およびClient #1は、通信インターフェースを共通にしているので、必ずしもオペレーティングシステムOSが同一である必要はない。

【0234】つぎに、図24に示した状態におけるServer #1およびClient #1のサービスモジュールデータについて説明する。ここでは特に、Client #1がServer #1に対して共通モジュールServiceModule #D2の実行依頼をおこない、さらにServer #1がServer #2に対して共通モジュールServiceModule #D2の実行依頼をおこなった場合を考える。

【0235】図25は、この場合のServer #1のサーバデータ管理テーブルSDT1に登録された共通モジュールServiceModule #D2のサービスモジュールデータを示す説明図である。

【0236】図25において、まず、「サービスモジュール名」として「ServiceModule #D2」が登録され、「デフォルトサービス名」として「Service #1」が登録されている。そして、「モジュール種別」として共通モジュールを示すコードである「4」が登録され、「プラットフォーム種別」として、サーバ、クライアント、ハードウェア仕様またはオペレーティングシステムを特定しないコード「0」が登録されている。

【0237】さらに、「版数」として「1」が登録され、「状態フラグ」としてリモート実行、すなわちServer #1以外のサーバによる共通モジュールServiceModule #D2の実行を示すコードである「2」が登録されている。そして、「リモート実行時のサーバアドレス」としてこの共通モジュールServiceModule #D2の起動をおこなうサーバであるServer #2のアドレスを示す「#0002」が登録され、「アクセスID」としてこのServer #2上の共通モジュールServiceModule #D2にアクセスするのに必要なID「333」が登録されている。

【0238】図26は、この場合のClient #1のクライアントデータ管理テーブルCDT1に登録された共通モジュールServiceModule #D1のサービスモジュールデータを示す説明図である。

【0239】図26において、まず、「サービスモジュール名」として「ServiceModule #D1」

50

が登録され、「デフォルトサービス名」として「Service #1」が登録されている。そして、「版数」として「1」が登録され、「取得サーバアドレス」としてこの共通モジュールServiceModule #D2を配信したサーバの論理アドレスを示す「#0001」が登録されている。

【0240】さらに、「状態フラグ」としてリモート実行、すなわちサーバによる共通モジュールServiceModule #D2の実行を示すコードである「2」が登録されている。そして、「リモート実行時のサーバアドレス」としてこの共通モジュールServiceModule #D2の起動をおこなうサーバであるServer #2のアドレスを示す「#0002」が登録され、「アクセスID」としてこのServer #2上の共通モジュールServiceModule #D2にアクセスするのに必要なID「333」が登録されている。

【0241】以上に説明したように、実施の形態2にかかる分散処理システムによれば、上述した「サーバにおけるモジュール実行環境移行処理」の実行によって、サーバの資源が圧迫された場合に、サーバにおいて実行されていた共通モジュールの一部を、リソースに余裕のある他のサーバ上で起動することができ、サーバのリソース不足によるスループットの低下を迅速に回避することができるとともに、効果的な負荷分散を実現することができる。

【0242】また、以上に説明した実施の形態1および2によれば、サーバからクライアントに配信されるサービスモジュールの属性（サービスモジュール名、デフォルトサービス名、版数等）や実行状態（状態フラグ）を登録設定されるサービスモジュールデータを備えたサーバデータ管理テーブルSDT1、クライアントデータ管理テーブルCDT1を、サーバ、クライアントにそれぞれ備えることで、サービスの実行に必要なサービスモジュールの実行位置が把握でき、上述した「サービスモジュールアクセス処理」を実行することによって、共通モジュールへのアクセスを確実におこなうことができる。

【0243】なお、以上に説明した原理説明、実施の形態1および2において、図6～15に示した処理を実行させるプログラム（詳しくは個別に実行可能なモジュール群）をICカードメモリ、フロッピーディスク、光磁気ディスク、CD-ROM等の記録媒体に格納し、この記録媒体に記録されたプログラムをサーバ装置およびクライアント装置にインストールすることで、上述した分散処理方法を使用し、これにより分散処理システムを構築することもできる。なお、このインストール作業は、通信回線を使用してサーバからクライアントに対して、またはサーバ間においてダウンロードすることによっておこなってもよい。

【0244】

50

51

【発明の効果】以上、説明したとおり、本発明の請求項1にかかる分散処理システムおよび請求項5にかかる分散処理方法によれば、サーバ装置は、クライアント装置の要求に応じて、クライアント装置の種別毎に異なり、かつサーバの実行に必要なモジュールをサーバ装置から取得することのできる初期設定モジュールをクライアント装置に配信し、クライアント装置は、サーバ装置からこの初期設定モジュールを取得して実行することで、クライアント装置において必要なサービスモジュールのみをサーバ装置から取得することができ、この取得したサービスモジュールを実行することにより、サービスを利用することができるので、ハードウェア仕様やオペレーティングシステムの異なるクライアント装置がネットワークに接続された場合であっても、クライアント装置に応じた初期設定モジュールを取得するだけで、サーバ装置によって提供されるサービスがクライアント装置間において差異なく利用でき、また、クライアント装置においてサービス利用手段により該当する初期設定モジュールを自動的に取得することができるので、ユーザは、クライアント装置の種別を意識することなく、サーバ装置の提供するサービスを容易に利用することができる。そして、本発明の請求項9にかかる記録媒体を用いれば、サーバ装置またはクライアント装置として用いられるコンピュータに対して、上記した請求項5にかかる分散処理方法を使用することができ、上記した請求項1にかかる分散処理システムの構築が可能となる。

【0245】また、本発明の請求項2にかかる分散処理システムおよび請求項6にかかる分散処理方法によれば、サーバ装置に、実行環境を問わない共通モジュールが備えられることで、サーバ装置のリソース不足状態に対して、本来、このサーバ装置において実行すべき共通モジュールを、他のサーバ装置において実行するように実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、多くのクライアント装置のアクセスにより、サーバ装置の資源が圧迫された場合に、サーバ装置において実行されていた共通モジュールを、リソースに余裕のある他のサーバ装置において実行できるようになり、サーバ装置のリソース不足によるスループットの低下を回避することができる。そして、本発明の請求項10にかかる記録媒体を用いれば、サーバ装置またはクライアント装置として用いられるコンピュータに対して、上記した請求項6にかかる分散処理方法を使用することができ、上記した請求項2にかかる分散処理システムの構築が可能となる。

【0246】また、本発明の請求項3にかかる分散処理システムおよび請求項7にかかる分散処理方法によれば、サーバ装置にサーバ装置とクライアント装置との双方の実行環境において実行可能な共通モジュールが備えられることで、クライアント装置のリソース不足状態に対して、本来、クライアント装置において実行すべき共

52

通モジュールを、サーバ装置において実行するように実行位置を移行することができるので、たとえば、サービスプログラムの実行中に、他のアプリケーションプログラム等の実行により、クライアント装置の資源が圧迫された場合に、クライアント装置において実行されていた共通モジュールをサーバ装置において実行できるようになり、クライアント装置のリソース不足によるスループットの低下を回避することができる。そして、本発明の請求項11にかかる記録媒体を用いれば、サーバ装置またはクライアント装置として用いられるコンピュータに対して、上記した請求項7にかかる分散処理方法を使用することができ、上記した請求項3にかかる分散処理システムの構築が可能となる。

【0247】また、本発明の請求項4にかかる分散処理システムおよび請求項8にかかる分散処理方法によれば、サーバ装置が、クライアント装置に対してサービスモジュールの更新を通知することができ、さらに、その通知は、過去にそのサービスモジュールの配信をおこなった先のクライアント装置に対しておこなわれるので、クライアント装置において、ユーザが、特にサーバ装置におけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能となる。そして、本発明の請求項12にかかる記録媒体を用いれば、サーバ装置またはクライアント装置として用いられるコンピュータに対して、上記した請求項8にかかる分散処理方法を使用することができ、上記した請求項4にかかる分散処理システムの構築が可能となる。

【0248】また、請求項13の発明によれば、クライアント装置に対して、クライアント装置の種別毎に異なるサービスモジュールの配信を可能とする初期設定モジュールを配信するので、クライアント装置側でサービスを利用するためのサービスプログラムに必要なサービスモジュールの変更等が生じた場合にも、初期設定モジュールの内容を変更するだけで対応でき、クライアント装置の種別毎に存在する多くのサービスモジュールの管理が容易なサーバ装置が得られるという効果を奏する。

【0249】また、請求項14の発明によれば、実行環境を問わない共通モジュールが備えられることで、リソース不足状態に対して、本来、このサーバ装置において実行すべき共通モジュールを、他のサーバ装置において実行するように実行位置を移行することができるので、リソース不足が生じた場合に、実行されていた共通モジュールを、リソースに余裕のある他のサーバ装置において実行できるようになり、これによりスループットの低下を回避することが可能なサーバ装置が得られるという効果を奏する。

【0250】また、請求項15の発明によれば、実行されている複数の共通モジュールのうち、リソース不足が解消されるだけの共通モジュールを、他のサーバ装置に対する起動要求の対象とするので、起動要求先のサーバ

50

装置のリソースへの大きな圧迫を回避することが可能なサーバ装置が得られるという効果を奏する。

【0251】また、請求項16の発明によれば、起動要求された共通モジュールへのアクセスを、起動要求を発生したサーバ装置だけでなく、その共通モジュールの実行を必要とするクライアント装置に対しても直接に可能とするので、起動要求された共通モジュールの利用を容易にするサーバ装置が得られるという効果を奏する。

【0252】また、請求項17の発明によれば、モジュール実行位置テーブルによって共通モジュールの実行位置を管理するので、このモジュール実行位置テーブルを参照することで、所望の共通モジュールのアクセスを容易にするサーバ装置が得られるという効果を奏する。

【0253】また、請求項18の発明によれば、モジュール配信先テーブルを参照することによって、更新されたサービスモジュールの取得が必要なクライアント装置のみに更新通知をおこなうので、効率的なサービスモジュールの更新処理が可能なサーバ装置が得られるという効果を奏する。

【0254】また、請求項19の発明によれば、サービス利用手段によって、サーバ装置から初期設定モジュールを取得して実行するだけで、この初期設定モジュールの処理により、サービスを利用するのに必要なサービスモジュールのみをサーバ装置から取得することができるので、サーバ装置によって提供されるサービスが差異なく利用できることと、ユーザが、クライアント装置の種別を意識することなく、サーバ装置の提供するサービスを容易に利用することができるクライアント装置が得られるという効果を奏する。

【0255】また、請求項20の発明によれば、サービス利用手段において、クライアント装置の機能と処理能力を調査する調査機構が設けられ、初期設定モジュールの実行時にこの調査機構を実行して取得される調査結果に応じて、サーバ装置から取得するサービスモジュールを選択することができるので、調査結果が処理能力の低下を示す場合に、実行可能なサービスモジュールだけを取得して実行し、サービス実行に必要な他のサービスモジュールの実行をサーバ装置等に依頼することが可能になり、これによりストレスのないサービス利用が可能なクライアント装置が得られるという効果を奏する。

【0256】また、請求項21によれば、クライアントデータ管理テーブルに、サービスを利用するのに必要なサービスモジュールが登録されていない場合に、そのサービスモジュールをサーバ装置から取得して、クライアントデータ管理テーブルに登録するので、サービスの利用に必要な最低限のサービスモジュールだけを付属する記憶装置等に備えておくことができ、資源の有効活用を実現することができるクライアント装置が得られるという効果を奏する。

【0257】また、請求項22によれば、実行環境を問

わない共通モジュールを実行している状態におけるリソース不足状態に対して、本来、このクライアント装置において実行すべき共通モジュールを、サーバ装置において実行するように実行位置を移行することができるのでリソース不足が生じた場合に、実行されていた共通モジュールを、リソースに余裕のあるサーバ装置において実行できるようになり、これによりスループットの低下を回避することができるクライアント装置が得られるという効果を奏する。

【0258】また、請求項23によれば、実行されている複数の共通モジュールのうち、リソース不足が解消されるだけの共通モジュールを、サーバ装置に対する起動要求の対象とするので、起動要求先のサーバ装置のリソースへの大きな圧迫を回避することができるクライアント装置が得られるという効果を奏する。

【0259】また、請求項24によれば、モジュール実行位置テーブルによって共通モジュールの実行位置を管理するので、このモジュール実行位置テーブルを参照することで、所望の共通モジュールのアクセスが容易なクライアント装置が得られるという効果を奏する。

【0260】また、請求項25によれば、更新モジュール取得手段によって、サーバ装置からサービスモジュールの更新が通知された際に、更新されたサービスモジュールを取得するので、ユーザ自らが、サーバ装置におけるモジュールの更新時期を意識する必要がなくなり、かつ効率的なサービスモジュールの更新処理が可能なクライアント装置が得られるという効果を奏する。

【図面の簡単な説明】

【図1】本発明にかかる分散処理システムにおいて、提供されるサービス毎に用意されたサービスデータのモジュール構成を示す説明図である。

【図2】本発明にかかるサーバのシステム実行形態を示す説明図である。

【図3】本発明にかかるサーバデータ管理テーブルの各サービスモジュールに対するサービスモジュールデータを示した説明図である。

【図4】本発明にかかるクライアントのシステム実行形態を示す説明図である。

【図5】本発明にかかるクライアントデータ管理テーブルの各サービスモジュールに対するサービスモジュールデータを示した説明図である。

【図6】本発明にかかるクライアントデモンストラの起動において実行される処理を示したフローチャートである。

【図7】本発明にかかるサーバデモンストラの起動時に実行される処理を示したフローチャートである。

【図8】本発明にかかるシステム監視スレッドによる処理を示したフローチャートである。

【図9】本発明にかかる初期設定モジュール取得処理を示したフローチャートである。

【図10】本発明にかかる初期設定モジュールによる処

理を示したフローチャートである。

【図 11】本発明にかかるサービス起動処理を示したフローチャートである。

【図 12】本発明にかかるクライアントにおけるモジュール実行環境移行処理を示したフローチャートである。

【図 13】本発明にかかるサービスモジュールアクセス処理を示したフローチャートである。

【図 14】本発明にかかるサーバにおけるモジュール実行環境移行処理を示したフローチャートである。

【図 15】本発明にかかるサービスモジュール更新処理 10

【図 16】実施の形態 1 にかかる分散処理システムのサービス未起動時のシステム構成を示した説明図である。

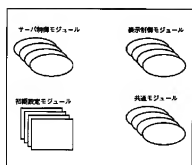
【図 17】実施の形態 1 において、Service # 1 を実行するために必要なサービスデータを示す説明図である。

【図 18】実施の形態 1 にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。

【図 19】実施の形態 1 において、Server # 1 の初期設定モジュール ServiceModule # C1 20

【図 20】実施の形態 1 において、Client # 1 の共通モジュール ServiceModule # D1 のサ

【図 1】



【図 5】

サービスモジュール名:	<input type="text"/>
デフォルトサービス名:	<input type="text"/>
種別:	<input type="text"/> 監視サーバアドレス: <input type="text"/>
状態フラグ:	<input type="text"/>
ローカル実行時のアクセスID:	<input type="text"/>
リモート実行時のサーバアドレス:	<input type="text"/> アクセスID: <input type="text"/>

ービスモジュールデータを示す説明図である。

【図 21】実施の形態 1 にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。

【図 22】実施の形態 1 において、Server # 1 の共通モジュール ServiceModule # D1 のサービスモジュールデータを示す説明図である。

【図 23】実施の形態 1 において、Client # 1 の共通モジュール ServiceModule # D1 のサービスモジュールデータを示す説明図である。

【図 24】実施の形態 2 にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。

【図 25】実施の形態 2 において、Server # 1 の共通モジュール ServiceModule # D1 のサービスモジュールデータを示す説明図である。

【図 26】実施の形態 2 において、Client # 1 の共通モジュール ServiceModule # D1 のサービスモジュールデータを示す説明図である。

【符号の説明】

SD1, SD2 サーバデーモン

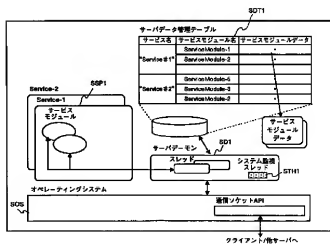
CD1 クライアントデーモン

SDT1, SDT2 サーバデータ管理テーブル

CDT1 クライアントデータ管理テーブル

STH1, STH2, CTH1 システム監視スレッド

【図 2】



【図 20】

サービスモジュール名:	ServiceModule # D1
デフォルトサービス名:	Service # 1
種別:	1 監視サーバアドレス: # 0001
状態フラグ:	1
ローカル実行時のアクセスID:	111
リモート実行時のサーバアドレス:	<input type="text"/> アクセスID: <input type="text"/>

【図3】

サービスモジュール名:

デフォルトサービス名:

配達先クライアントアドレスリスト:

モジュール種類:

プラットフォーム種別:

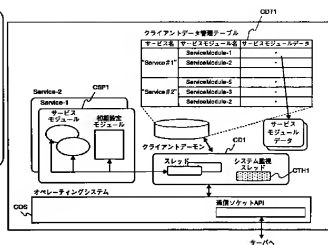
種別:

拡張プラグ:

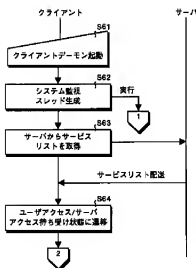
ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

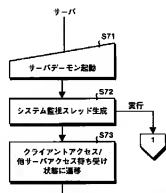
【図4】



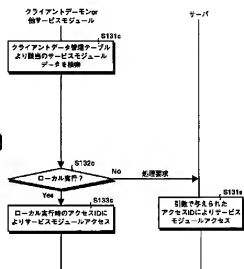
【図6】



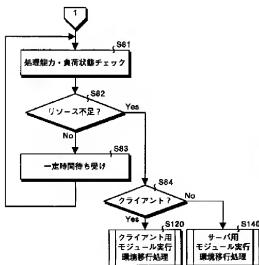
【図7】



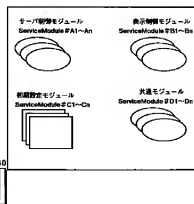
【図13】



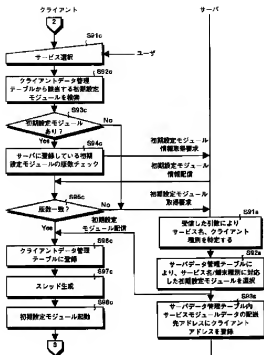
【図8】



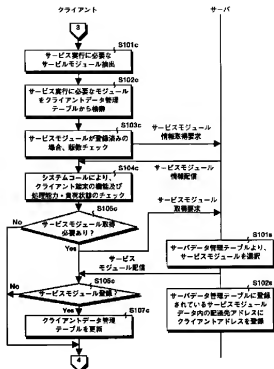
【図17】



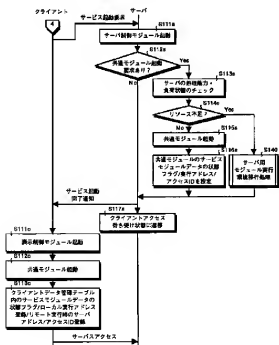
【图9】



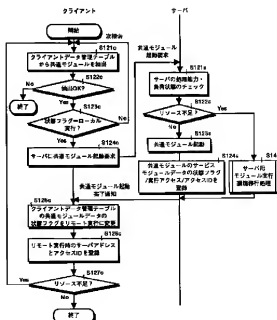
【図10】



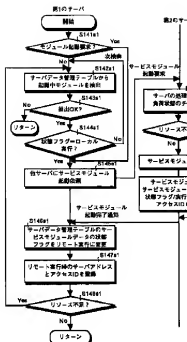
【图 1-1】



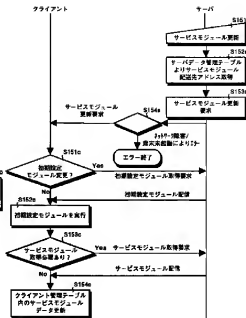
【图 12】



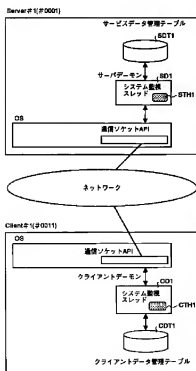
【图 1-4】



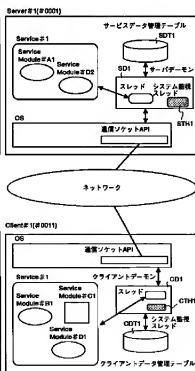
【图 15】



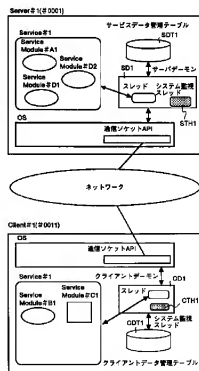
【图 1-6】



【图 18】



【图 2-1】



【図19】

サービスモジュール名:

デフォルトサービス名:

配属先クライアントアドレスリスト:

モジュール種類:

プラットフォーム種類:

版数:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【図22】

サービスモジュール名:

デフォルトサービス名:

配属先クライアントアドレスリスト:

モジュール種類:

プラットフォーム種類:

版数:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【図23】

サービスモジュール名:

デフォルトサービス名:

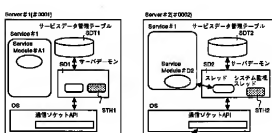
版数: 状態サーバアドレス:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【図24】



【図25】

サービスモジュール名:

デフォルトサービス名:

配属先クライアントアドレスリスト:

モジュール種類:

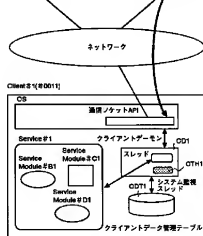
プラットフォーム種類:

版数:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:



【図26】

サービスモジュール名:

デフォルトサービス名:

版数: 状態サーバアドレス:

状態フラグ:

ローカル実行時のアクセスID:

リモート実行時のサーバアドレス: アクセスID:

【手続補正書】

【提出日】平成 11 年 8 月 20 日（1999. 8. 20）

【手続補正 1】

【補正対象書類名】明細書

【補正対象項目名】請求項 4

【補正方法】変更

【補正内容】

【請求項 4】 前記サーバ装置は、前記モジュールが更新された際に、該モジュールを配信した先のクライアント装置に、前記更新されたモジュールの更新を通知する更新通知手段を備え、前記クライアント装置は、前記更新されたモジュールの更新が通知された場合に、前記サーバ装置から前記更新されたモジュールを取得する更新モジュール取得手段を備えたことを特徴とする請求項 1～3 のいずれか一つに記載の分散処理システム。

【手続補正 2】

【補正対象書類名】明細書

【補正対象項目名】請求項 10

【補正方法】変更

【補正内容】

【請求項 10】 ネットワークを介して互いに接続されたサーバ装置とクライアント装置とを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、前記サーバ装置と他のサーバ装置との双方の実行環境において実行可能な共通モジュールを前記サーバ装置において実行するサーバ共通モジュール実行ステップと、前記サーバ装置が処理能力の低下や負荷の増大等によるリソース不足状態となった場合に、他のサーバ装置に対して該他のサーバ装置側での前記共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップと、を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【手続補正 3】

【補正対象書類名】明細書

【補正対象項目名】0044

【補正方法】変更

【補正内容】

【0044】 また、請求項 10 の発明において、ネットワークを介して互いに接続されたサーバとクライアントとを備えた分散処理システムにおける分散処理方法をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体は、サーバと他のサーバとの双方の実行環境において実行可能な共通モジュールをサーバにおいて実行するサーバ共通モジュール実行ステップ（「サービス起動処理」のステップ S115s、「クライアントにおけるモジュール実行環境移行処理」

のステップ S123s、または「サーバにおけるモジュール実行環境移行処理」のステップ S143s に相当）と、サーバが処理能力の低下や負荷の増大等によるリソース不足状態となった場合（「システム監視スレッドの処理」に相当）に、他のサーバに対してその他のサーバ側での共通モジュールの起動を要求するサーバ共通モジュール起動要求ステップ（「サーバにおけるモジュール実行環境移行処理」のステップ S145s1 に相当）と、を実行させることを特徴とする。

【手続補正 4】

【補正対象書類名】明細書

【補正対象項目名】0206

【補正方法】変更

【補正内容】

【0206】 図 18 は、実施の形態 1 にかかる分散処理システムのサービス起動後のシステム構成を示した説明図である。図 18 において、Server #1 は、図 17 に示す状態に加えて、Service #1 を提供するサーバ用サービスプログラムとして、サーバ制御モジュール ServiceModule #A1、共通モジュール ServiceModule #D2 が用意され、これらサービスモジュールの実行においてサーバデーモン SD1 内にスレッドが生成される。

【手続補正 5】

【補正対象書類名】明細書

【補正対象項目名】0207

【補正方法】変更

【補正内容】

【0207】 また、Client #1 は、図 17 に示す状態に加えて、Service #1 を提供するクライアント用サービスプログラムとして、表示制御モジュール ServiceModule #B1、初期設定モジュール ServiceModule #C1、共通モジュール ServiceModule #D1 が用意され、これらサービスモジュールの実行においてクライアントデーモン CD1 内にスレッドが生成される。

【手続補正 6】

【補正対象書類名】明細書

【補正対象項目名】0218

【補正方法】変更

【補正内容】

【0218】 図 22 において、まず、「サービスモジュール名」として「ServiceModule #D1」が登録され、「デフォルトサービス名」として「Service #1」が登録されている。そして、「配送先クライアントリスト」には、上記した Client #1 の論理アドレスである「#0011」が登録されている。

【手続補正 7】

【補正対象書類名】明細書

